PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/66 — A1

(11) International Publication Number: WO 99/65198

(43) International Publication Date: 16 December 1999 (16.12.99)

(21) International Application Number: PCT/US99/13311

(22) International Filing Date: 11 June 1999 (11.06.99)

(30) Priority Data:

60/088,891	11 June 1998 (11.06.98)	US
60/088,893	11 June 1998 (11.06.98)	US
60/088,906	11 June 1998 (11.06.98)	US
60/088,914	11 June 1998 (11.06.98)	US
60/088,915	11 June 1998 (11.06.98)	US
60/088,983	11 June 1998 (11.06.98)	US
09/120,515	22 July 1998 (22.07.98)	US
09/120,529	22 July 1998 (22.07.98)	US
09/120,636	22 July 1998 (22.07.98)	US
09/120,672	22 July 1998 (22.07.98)	US
09/120,700	22 July 1998 (22.07.98)	US
09/120,944	22 July 1998 (22.07.98)	US

(71) Applicant: SYNCHRODYNE INC. [US/US]; 75 Maiden Lane, New York, NY 10038 (US).

(72) Inventors: OFEK, Yoram; Suite 1921, 2600 Netherland Avenue, Riverdale, NY 10463 (US). SHACHAM, Nachum; 1146 Stanley Way, Palo Alto, CA 94303 (US).

(74) Agent: SITRICK, David, H.; Sitrick & Sitrick, Suite 201, 8340 N. Lincoln Avenue, Skokie, IL 60077 (US).

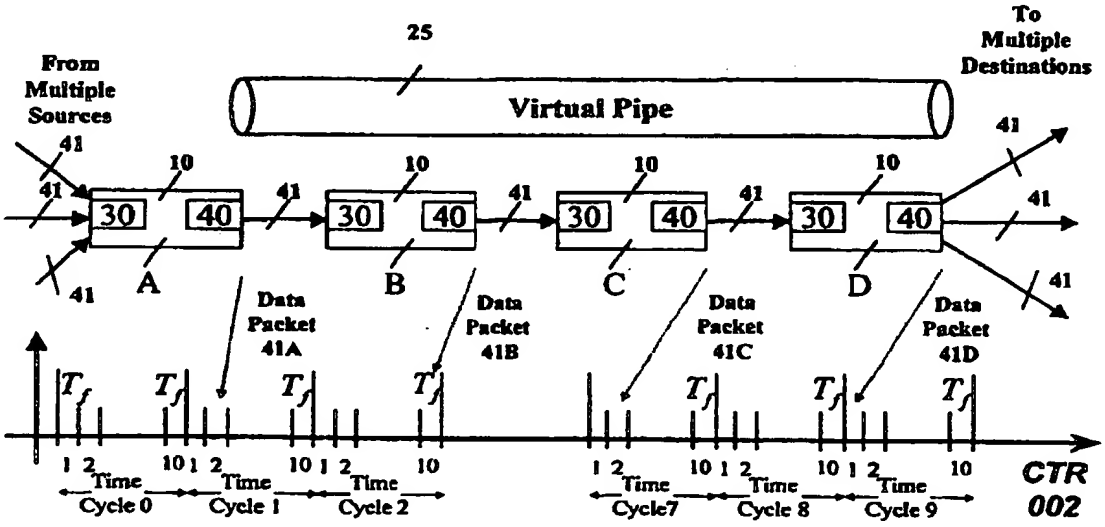
(81) Designated States: CA, CN, JP, KP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: COMMON TIME REFERENCE FOR PACKET SWITCHES



(57) Abstract

The invention describes a method for transmitting and forwarding packets over a packet switching network via communication links with variable delays. The switches (10) of the network maintain a common time reference (002), which is obtained either from an external source or is generated and distributed internally. A packet that arrives to an input port (30) of a switch (10) is switched to an output port (40) based on specific routing information in the packet header. Each switch (10) along a route form a source to a destination forwards packets in periodic time intervals that are predefined using the common time reference. The time interval duration can be longer than the time duration required for transmitting a packet, in which case the exact position of a packet in the time interval is not predetermined.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

COMMON TIME REFERENCE FOR PACKET SWITCHES

BACKGROUND OF THE INVENTION:

This invention relates generally to a method and apparatus for transmitting of data on a communications network. More specifically, this invention relates to timely forwarding and delivery of data over the network and to their destination nodes. The timely forwarding is possible by time information that is globally available from a global positioning system (GPS). Consequently, the end-to-end performance parameters, such as, loss, delay and jitter, have either deterministic or probablistic guarantees.

This invention further facilitates a method and apparatus for integrating the transfer of two traffic types over a data packet communications network. More specifically, this invention provides timely forwarding and delivery of data packets from sources with constant bit rate (CBR) and variable bit rate (VBR) over the network and to their destination nodes.

This invention further relates a method and apparatus for transmitting of data on a communications network via communication links with variable delays. More specifically, this invention relates to timely forwarding and delivery of data over networks with links with variable delays to their destination nodes, while ensuring the end-to-end performance parameters, such as, loss, delay and jitter, have either deterministic or probablistic guarantees.

This invention also provides a method and apparatus for monitoring, policing, and billing of the transmission of data packet on a communications network. More specifically, this invention provides the monitoring, policing, and billing in networks with timely forwarding and delivery of data packets to their destination nodes. Consequently, the end-to-end performance parameters, such as, loss, delay and jitter, are predictable, and therefore, it is possible to measure them. Consequently, such measurements are used in the monitoring, policing and billing.

This invention also provides for a method and apparatus for transmitting of data on an heterogeneous communications network, which has two types of switching nodes: (i) asynchronous and (ii) synchronous with common time reference. More specifically, this invention provides timely forwarding and delivery of data over the network and to their destination nodes. Consequently, the end-to-end performance parameters, such as, loss, delay and jitter, have either deterministic or probablistic guarantees.

This invention furthermore facilitates a routing decision by using both timing information and position information of packets within time frames. In this case, there

is no need to decode the address in the packet header, it is feasible to encrypt the entire data packet (including the header) as it is transferred through a public backbone network, which is an important security feature. Consequently, over this novel communications network it is possible to transport wide variety of data packets, such as, IP (Internet protocol) and ATM (asynchronous transfer mode).

The proliferation of high-speed communications links, fast processors, and affordable, multimedia-ready personal computers brings about the need for wide area networks that can carry real time data, like telephony and video. However, the end-to-end transport requirements of real-time multimedia applications present a major challenge that cannot be solved satisfactorily by current networking technologies. Such applications as video teleconferencing, and audio and video multicasting generate data at a wide range of bit rates and require predictable, stable performance and strict limits on loss rates, average delay, and delay variations ("jitter"). These characteristics and performance requirements are incompatible with the services that current circuit and packet switching networks can offer.

Packet switching networks like IP (Internet Protocol)-based Internet and Intranets [see, for example, A.Tannebaum, "Computer Networks" (3rd Ed) Prentice Hall, 1996] and ATM (Asynchronous Transfer Mode) [see, for example, Handel et al., "ATM Networks: Concepts, Protocols, and Applications", (2nd Ed.) Addison-Wesley, 1994] handle bursty data more efficiently than circuit switching, due to their statistical multiplexing of the packet streams. However, current packet switches and routers operate asynchronously and provide best effort service only, in which end-to-end delay and jitter are neither guaranteed nor bounded. Furthermore, statistical variations of traffic intensity often lead to congestion that results in excessive delays and loss of packets, thereby significantly reducing the fidelity of real-time streams at their points of reception. In fact, under best effort service, the performance characteristics of a given connection are not even predictable at the time of connection establishment.

Efforts to define advanced services for both IP and ATM have been conducted in two levels: (1) definition of service, and (2) specification of methods for providing different services to different packet streams. The former defines interfaces, data formats, and performance objectives. The latter specifies procedures for processing packets by hosts and switches/routers. The types of services that defined for ATM include constant bit rate (CBR), variable bit rate (VBR) and available bit rate (ABR). For IP, the defined services include guaranteed performance (bit rate, delay), controlled flow, and best effort [J.Wroclawski, "Specification of the Controlled-Load Network Element Service", IETF RFC 2211, September 1997] [Shenker et. al., "Specification of Guaranteed Quality of Service", IETF RFC 2212. September 1997]. Signaling

protocols, e.g., RSVP and UNI3.1, which carry control information to facilitate the establishment of the desired services, are specified for IP and ATM, respectively [R. Braden, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, IETF Request for Comment RFC2205", September 1997] [Handel et al., "ATM Networks: Concepts, Protocols, and Applications", (2nd Ed.) Addison-Wesley, 1994]. These protocols address the transport of data to one destination known as unicast or multiple destinations multicast. In addition, SIP, a higher level protocol for facilitating the establishment of sessions that use the underlying services, is currently under definition under IETF auspices [Handley et al., "SIP-Session Initiation Protocol", <draft-draft-ietf-mmusic-sip-04.ps>, November 1997].

The methods for providing different services under packet switching fall under the general title of Quality of Service (QoS). Prior art in QoS can be divided into two parts: (1) traffic shaping with local timing without deadline scheduling, for example [M.G.H. Katevenis, "Fast Switching And Fair Control Of Congested Flow In Broadband Networks", IEEE Journal on Selected Areas in Communications, SAC-5(8):1315-1326, October 1987; Demers et al., "Analysis and Simulation Of A Fair Queuing Algorithm", ACM Computer Communication Review (SIGCOMM'89), pages 3-12, 1989; S.J. Golestani, "Congestion-Free Communication In High-Speed Packet Networks", IEEE Transcripts on Communications, COM-39(12):1802-1812, December 1991; Parekh et al., "A Generalized Processor Sharing Approach To Flow Control - The Multiple Node Case", IEEE/ACM T. on Networking, 2(2):137-150, 1994], and (2) traffic shaping with deadline scheduling, for example [Ferrari et al., "A Scheme For real-time Channel Establishment In Wide-Area Networks", IEEE Journal on Selected Areas in Communication, SAC-8(4):368-379, April 1990; Kandlur et al., "Real Time Communication In Multi-Hop Networks", IEEE Trans. on Parallel and Distributed Systems, Vol. 5, No. 10, pp. 1044-1056, 1994]. Both of these approaches rely on manipulation of local queues by each router with little coordination with other routers. The Weighted Fair Queuing (WFQ), which typifies these approaches, is based on cyclical servicing of the output port queues where the service level of a specific class of packets is determined by the amount of time its queue is served each cycle [Demers et al., "Analysis and Simulation Of A Fair Queuing Algorithm", ACM Computer Communication Review (SIGCOMM'89), pages 3-12, 1989]. These approaches have inherent limitations when used to transport real-time streams. When traffic shaping without deadline scheduling is configured to operate at high utilization with no loss, the delay and jitter are inversely proportional to the connection bandwidth, which means that low rate connections may experience large delay and jitter inside the network. In traffic

shaping with deadline scheduling the delay and jitter are controlled at the expense of possible congestion and loss.

The recognition that the processing of packets by switches and routers constitutes a performance bottleneck resulted in the development of methods for enhancing performance by simplifying the processing of packets. Multiprotocol Label Switching (MPLS) converts the destination address in the packet header into a short tag, which defines the routing of the packet inside the network [Callon et al., "A Proposed Architecture For MPLS" <draft-ietf-mpls-arch-00.txt> INTERNET DRAFT, August 1997].

The real-time transport protocol (RTP) [H. Schultzrinne et. al, RTP: A Transport Protocol for Real-Time Applications, IETF Request for Comment RFC1889, January 1996] is a method for encapsulating time-sensitive data packets and attaching to the data time related information like time stamps and packet sequence number. RTP is currently the accepted method for transporting real time streams over IP internetworks and packet audio/video telephony based on ITU-T H.323.

Synchronous methods are found mostly in circuit switching, as compared to packet switching that uses mostly asynchronous methods. However, some packet switching synchronous methods have been proposed. IsoEthernet or IEEE 802.9a [IEEE 802.9a Editor. Integrated service (is): IEEE 802.9a "Isochronous Services With CSMA/CD MAC Service", IEEE Draft, March 1995] combines CSMA/CD (IEEE 802.3), which is an asynchronous packet switching, with N-ISDN and H.320, which is circuit switching, over existing Ethernet infrastructure (10Base-T). This is a hybrid solution with two distinct switching methods: N-ISDN circuit switching and Ethernet packet switching. The two methods are separated in the time domain by time division multiplexing (TDM). The IsoEthernet TDM uses fixed allocation of bandwidth for the two methods - regardless of their utilization levels. This approach to resource partitioning results in undesirable side effects like under-utilization of the circuit switching part while the asynchronous packet switching is over loaded but cannot use the idle resources in the circuit switching part.

One approach to an optical network that uses synchronization was introduced in the synchronous optical hypergraph [Y. Ofek, "The Topology, Algorithms And Analysis Of A Synchronous Optical Hypergraph Architecture", Ph.D. Dissertation, Electrical Engineering Department, University of Illinois at Urbana, Report No. UIUCDCS-R-87-1343, May 1987], which also relates to how to integrate packet telephony using synchronization [Y. Ofek, "Integration Of Voice Communication On A Synchronous Optical Hypergraph", INFOCOM'88, 1988]. In the synchronous optical hypergraph, the forwarding is performed over hyper-edges, which are passive optical stars. In [Li et

al., "Pseudo-Isochronous Cell Switching In ATM Networks", IEEE INFOCOM'94, pages 428-437, 1994; Li et al., "Time-Driven Priority: Flow Control For Real-Time Heterogeneous Internetworking", IEEE INFOCOM'96, 1996] the synchronous optical hypergraph idea was applied to networks with an arbitrary topology and with point-to-point links. The two papers [Li et al., "Pseudo-Isochronous Cell Switching In ATM Networks", IEEE INFOCOM'94, pages 428-437, 1994; Li et al., "Time-Driven Priority: Flow Control For Real-Time Heterogeneous Internetworking", IEEE INFOCOM'96, 1996] provide an abstract (high level) description of what is called "RISC-like forwarding", in which a packet is forwarded, with little if any details, one hop every time frame in a manner similar to the execution of instructions in a Reduced Instruction Set Computer (RISC) machine.

In U.S. Pat. No. 5,418,779 Yemini et al. discloses switched network architecture with common time reference. The time reference is used in order to determine the time in which multiplicity of nodes can transmit simultaneously over one predefined routing tree to one destination. At every time instance the multiplicity of nodes are transmitting to different single destination node.

Routing—the selection of an output port for an information segment (i.e. data packets) that arrives at an input port of a switch—is a fundamental function of communication networks. In circuit switching networks, the unit of switching is a byte, and the switching is made based on the location of the byte in a time frame. Establishing a connection in a circuit switching network requires the network to reserve a slot for the connection in every frame. The position of the byte in the frame is different from link to link, so each switch maintains a translation table from incoming frame positions on each input port to respective output ports and frame positions therein. The sequence of frame positions on the links of the route constitute a circuit that is assigned for the exclusive use of a specific connection, which results in significant inflexibility: the connection is limited in traffic intensity by the capacity of the circuit and when the connection does not use the circuit no other is allowed to use it. This feature is useful for CBR traffic, like PCM telephony, but it results in low utilization of the network when the traffic is bursty [C. Huitema, Routing in the Internet, Prentice Hall, 1995, and A. Tannebaum Computer Networks (3rd Ed) Prentice Hall 1996].

SUMMARY OF THE INVENTION:

In accordance with the present invention, a method is disclosed providing virtual pipes that carry real-time traffic over packet switching networks while guaranteeing end-to-end performance. The method combines the advantages of both circuit and packet

switching. It provides for allocation for the exclusive use of predefined connections and for those connections it guarantees loss free transport with low delay and jitter. When predefined connections do not use their allocated resources, other non-reserved data packets can use them without affecting the performance of the predefined connections. On the Internet the non-reserved data packet traffic is called "best effort" traffic.

This invention further describes a method for transmitting and forwarding packets over a packet switching network where the delay between two switches increases, decreases, or changes arbitrarily over time. Packets are being forwarded over each link inside the network in predefined periodic time intervals. The switches of the network maintain a common time reference, which is obtained either from an external source (such as GPS - Global Positioning System) or is generated and distributed internally. The time intervals are arranged with simple periodicity and complex periodicity (like seconds and minutes of a clock).

The invention provides methods for maintaining timely forwarding within predefined time interval over two types of link delay variations: (i) increasing delay and (ii) decreasing delay. When the delay increases at some point of time a packet may be late for its predefined forwarding time interval. In such case the packet is delayed until the next time interval of its virtual pipe. When the link delay decreases, packets are buffered until the first time interval of its virtual pipe.

Packets that are forwarded inside the network over the same route and in the same time intervals constitute a virtual pipe and share a pipe-ID. The pipe-ID can be either explicit, such as a tag or a label that is generated inside the network, or implicit, such as a group of IP addresses. A virtual pipe provides deterministic quality of service guarantees. The time interval in which a switch forwards a specific packet is determined by the packet's pipe-ID, the time it reaches the switch, and the current value of the common time reference.

In accordance with the present invention, the bandwidth allocated to a connection and the delay and jitter inside the network are independent. MPLS can be used by the present invention to identify virtual pipes. The packet time-stamp that is carried in the RTP header can be used in accordance with the present invention to facilitate time-based transport.

Under the aforementioned prior art methods for providing packet switching services, switches and routers operate asynchronously. The present invention provides real-time services by synchronous methods that utilize a time reference that is common to the switches and end stations comprising a wide area network. The common time reference can be realized by using UTC (Coordinated Universal Time), which is globally available via, for example, GPS (Global Positioning System - see, for example:

http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps.html). By international agreement, UTC is the same all over the world. UTC is the scientific name for what is commonly called GMT (Greenwich Mean Time), the time at the 0 (root) line of longitude at Greenwich, England. In 1967, an international agreement established the length of a second as the duration of 9,192,631,770 oscillations of the cesium atom. The adoption of the atomic second led to the coordination of clocks around the world and the establishment of UTC in 1972. The Time and Frequency Division of the National Institute of Standards and Technologies (NIST) (see http://www.boulder.nist.gov/timefreq) is responsible for coordinating with the International Bureau of Weights and Measures (BIPM) in Paris in maintaining UTC.

UTC timing is readily available to individual PCs through GPS cards. For example, TrueTime, Inc.'s (Santa Rosa, CA) PCI-SG provides precise time, with zero latency, to computers that have PCI extension slots. Another way by which UTC can be provided over a network is by using the Network Time Protocol (NTP) [D. Mills, "Network Time Protocol" (version 3) IETF RFC 1305]. However, the clock accuracy of NTP is not adequate for inter-switch coordination, on which this invention is based.

Although the present invention relies on time to control the flow of packets inside the network in a similar fashion as in circuit switching, there are major differences between the two approaches. In circuit switching, for each data unit (e.g., a byte) at the time it has been transmitted from its source, it is possible to predict deterministically the future times it will be transmitted from any switch along its route [Ballart et al., "SONET: Now It's The Standard Optical Network", IEEE Communications Magazine, Vol. 29 No. 3, March 1989, pages 8-15]. The time resolution of this advanced knowledge is much shorter than the data unit transmission time. On the other hand, in accordance with the present invention, for each data unit (e.g., a cell) at the time it has been transmitted from its source, it is possible to know the future time frames that this data unit will be forwarded along its route. However, the time frame, which constitutes the accuracy of this advance timing knowledge, is much larger than one data unit transmission time. For example, the transmission time of an ATM cell (53 bytes) over a gigabit per second link is 424 nanoseconds, which is 294 times smaller than a typical time frame of 125 microseconds used in one embodiment of the present invention. There are several consequences that further distinguish the present invention from circuit switching.

In accordance with the present invention, the use of reserved resources is allowed by all packet traffic whenever the reserved resources are not in use.

In accordance with the present invention, the synchronization requirements are independent of the physical link transmission speed, while in circuit switching the synchronization becomes more and more difficult as the link speed increases.

In accordance with the present invention, timing information is not used for routing, and therefore, as in the Internet, for example, the routing is done using IP addresses or a tag/label.

In accordance with the present invention, the Internet "best effort" packet forwarding strategy can be integrated into the system.

In accordance with this invention, a method is disclosed for monitoring and policing the packet traffic in a packet switching network where the switches maintain a common time reference.

In accordance with this invention, a designated points inside the network is enabled to ascertain the level of packet traffic in predefine time intervals, and control the flow of packets and bring it back to predetermined levels in cases where the traffic volume exceeds predetermined levels. The information collected by the designated points facilitates billing for Internet services based on network usage, and identification of faulty conditions and malicious forwarding of packets that cause excessive delay beyond predetermined value.

In accordance with this invention, a method is described for exchanging timing messages and data packets between synchronous switches with a common time reference, and between end-stations/gateways and other synchronous switches, over an asynchronous network, i.e., a network with asynchronous switches. The method entails transmission of messages conveying the common time reference to end-station/gateways that have no direct access to the common time reference, and data packets that are sent responsive to the timing information and predetermined scheduled time intervals.

In accordance with this invention, methods are provided for maintaining timely forwarding within predefined time interval over three network configurations: (i) end-station to synchronous switch over asynchronous (LAN) switches, (ii) end-station to gateway over asynchronous (LAN) switches and then to synchronous switch, (iii) synchronous switch to synchronous switch over asynchronous switches.

These and other aspects and attributes of the present invention will be discussed with reference to the following drawings and accompanying specification.

BRIEF DESCRIPTION OF THE DRAWINGS:

FIG. 1 is a schematic illustration of a virtual pipe and its timing relationship with a common time reference (CTR), wherein delay is determined by the number of time frames between the forward time out at Node A and the forward time out at Node D;

FIG. 2 is a schematic illustration of multiple virtual pipes sharing certain ones of the switches;

FIG. 3 is a schematic block diagram illustration of a switch that uses a common time reference from the GPS (Global Positioning System) for the timely forwarding of packets disclosed in accordance with the present invention;

FIG. 4 illustrates the relationship among the local common time reference (CTR) on the switches, and how the multiplicity of local times is projected on the real-time axis, wherein time is divided into time frames of a predefined duration;

FIG. 5 is a schematic illustration of how the common time reference is organized into contiguous time-cycles of *k* time-frames each and contiguous super-cycle of *l* time-cycles each;

FIG. 6 is a schematic illustration of the relationship of the network common time reference and UTC (Coordinated Universal Time), such that, each time-cycle has 100 time-frames, of 125 microseconds each, and 80 time-cycles are grouped into one super-cycle of one second;

FIG. 7 is a schematic illustration of a data packet pipeline as in FIG. 1, and correlating to data packet movement through the switches 10 versus time for forwarding over a virtual pipe with common time reference (CTR);

FIG. 8 illustrates the mapping of the time frames into and out of a node on a virtual pipe, wherein the mapping repeats itself in every time cycle illustrating time in versus forwarding time out;

FIG. 9 is an illustration of a serial transmitter and a serial receiver;

FIG. 10 is a table of the 4B/5B encoding scheme for data such as is used by the AM7968 - TAXI chip set in accordance with one embodiment of the present invention;

FIG. 11 is a table of the 4B/5B encoding scheme for control signals, such as, the time frame delimiter (TFD) such as is used by the AM7968, in accordance with one embodiment of the present invention;

FIG. 12 is a schematic block diagram of an input port with a routing controller;

FIG. 13 is a schematic diagram of the routing controller which determines to which output port an incoming data packet should be switched to and attaches the time of arrival (ToA) information to the data packet header;

FIG. 14 is a flow diagram of the routing controller operation;

FIGS. 15A and 15B illustrate two generic data packet headers with virtual pipe ID (PID), and priority bit (P), wherein FIG. 15A illustrates a packet without time-stamp field, and wherein FIG. 15B illustrates a packet with time-stamp field, and also shows how the common time-reference value, time of arrival (ToA), is attached by the routing controller;

FIG. 16 is a schematic block diagram of an output port with a scheduling controller and a serial transmitter;

FIG. 17 is a schematic block diagram of the double-buffer scheduling controller;

FIG. 18 is a flow diagram of the double-buffer scheduling controller 46 operation;

FIG. 19 is a functional block diagram of the general scheduling controller with its transmit buffer and select buffer controller;

FIG. 20 is a flow diagram describing the packet scheduling controller operation for computing the forwarding time of a packet based on the following input parameters: PID 35C, ToA 35T and the CTR 002;

FIG. 21 is a flow diagram illustrating the operation of the Select Buffer Controller 45D;

FIG. 22 illustrates the real-time protocol (RTP) packet header with time-stamp field of 32 bits; and

FIG. 23 is a flow diagram describing the packet scheduling controller operation for computing the dispatching-time of a packet based on the following input parameters: PID, ToA, CTR and the RTP time-stamp.

FIG. 24 is a schematic description of a switch with a common time reference partition into time-frames with predefined positions such that the input port can unambiguously identify the positions;

FIG. 25 is a description of the timing partition of the common time reference into cycle with *k* time frames in each, while each time frame is further partitioned into four predefined parts: a, b, c and d;

FIG. 26 is a schematic diagram of the time-based routing controller. This unit determines to which output port a data packet should be switched and attaches the time in and position information to the data packet header;

FIG. 27 is an example of a routing and scheduling table on one of the incoming input ports using the incoming time or time-frame of arrival (ToA) and the position counter value for determining: (i) the output port, (ii) the out-going time-frame, and (iii) the position of the out-going data packet within the out-going time-frame;

FIG. 28 is a schematic illustration of a data packet which is sent across the fabric to the output port;

FIG. 29 is an example of a routing and scheduling table on one of the incoming input ports using the time stamp and position information for determining: (i) the output port, (ii) the out-going time-frame, and (iii) the position of the out-going data packet within the out-going time-frame;

FIG. 30 is a flow diagram of the routing controller operation;

FIG. 31 is a flow diagram of the data packet scheduling controller 45A operation;

FIG. 32 is a flow diagram of a different embodiment of the Select Buffer Controller 45D;

FIG. 33 is a schematic diagram of another alternate embodiment of the routing controller which determines to which output port an incoming data packet should be switched to and attaches the time of arrival (ToA) information to the data packet header;

FIGS. 34A and 34B are schematic illustrations of two generic data packet headers with virtual pipe ID (PID) and priority bits (P1/P2): (A) a packet without a time-stamp field and (B) a packet with a time-stamp field. This drawing also shows how the common time-reference value, time of arrival (ToA), is attached by the routing controller;

FIG. 35 is a table classifying the data packets;

FIG. 36 is a flow diagram of an alternate embodiment of the routing controller operation;

FIG. 37 is a schematic diagram of the scheduling and congestion controller, where each buffer is divided into two parts, one for constant bit rate (CBR) and the other for variable bit rate (VBR);

FIG. 38 is a flow diagram describing an alternate embodiment of the packet scheduling and rescheduling controller operation for computing the forwarding time of a packet based on the following input parameters: pipe-ID 35C, Time of arrival 35T and the common time reference 002;

FIG. 39 is a flow diagram describing an alternate embodiment of the select buffer and congestion controller 45D;

FIG. 40 is a schematic illustration of a virtual pipe with a delay which varies in time between Node B and Node C;

FIG. 41 is a timing diagram of a virtual pipe with an increasing delay between Nodes B and C;

FIG. 42 is a timing diagram of a virtual pipe with a decreasing delay between Nodes B and C;

FIG. 43 is a schematic illustration of a virtual pipe, p, with an alternate virtual pipe, p';

FIG. 44 describes a node, Node E, in which two virtual pipes, p and p' are converging;

FIG. 45 is a schematic illustration of a resynchronization mechanism of two virtual pipes Pipe-ID = p and Pipe-ID = p' on Node E;

FIG. 46 is a schematic illustration of the delay analysis and scheduling controller with its transmit buffer and select buffer controller;

FIG. 47 is a flow diagram describing an alternate embodiment of the Select Buffer Controller;

FIG. 48 is a flow diagram describing the delay analysis and scheduling controller operation for computing the forwarding time of a data packet;

FIG. 49 specifies a program executed by the delay analysis and scheduler controller for mobile nodes with increasing and decreasing delays in their incoming links;

FIG. 50 specifies a program executed by the delay analysis and scheduler controller for communication links in which their delay can change instantly, such as it is the case for SONET links in a self-healing SONET rings; and

FIG. 51 specifies a program executed by the delay analysis and scheduler controller for combining two alternate paths p and p' into one path.

FIG. 52 is a schematic illustration of the delay monitoring controller;

FIG. 53 is a flow chart of the program executed by the delay monitoring controller;

FIG. 54 is a schematic illustration of the policing and load controller;

FIG. 55 is a flow chart of the program executed by the policing and load controller;

FIG. 56 is a schematic illustration of connection between an end-station and a synchronous virtual pipe switch that are separated by asynchronous LAN switches;

FIG. 57 is a schematic illustration of connections between end-stations and synchronous virtual pipe switches which are separated by some asynchronous switches and a gateway, which resynchronize the data packets before it is forwarded to the synchronous switch;

FIG. 58 is a schematic illustration of connection between two segments of virtual pipe switches which are separated by asynchronous switches and routers;

FIG. 59 is a diagram of the temporal relationship between the common time reference signals from a synchronous switch and the timely forwarding of packets from an end-station;

FIG. 60 is an illustration of the possible time of arrival (ToA) variations to a synchronous switch of data packets, which are forwarded over asynchronous switches;

FIG. 61 is a schematic illustration of a data packet resynchronization mechanism at the input port; and

FIG. 62 specifies a program executed by the delay analysis and scheduler controller for the case of finding the schedule exactly on the delay bound by using the time-stamp at the data packet header.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS:

While this invention is susceptible of embodiment in many different forms, there is shown in the drawing, and will be described herein in detail, specific embodiments thereof with the understanding that the present disclosure is to be considered as an exemplification of the principles of the invention and is not intended to limit the invention to the specific embodiments illustrated.

The present invention relates to a system and method for transmitting and forwarding packets over a packet switching network. The switches of the network maintain a common time reference, which is obtained either from an external source (such as GPS - Global Positioning System) or is generated and distributed internally. The time intervals are arranged in simple periodicity and complex periodicity (like seconds and minutes of a clock). A packet that arrives to an input port of a switch, is switched to an output port based on specific routing information in the packet's header (e.g., IPv4 destination address in the Internet, VCI/VPI labels in ATM). Each switch along a route from a source to a destination forwards packets in periodic time intervals that are predefined using the common time reference. The time interval duration can be longer than the time duration required for transmitting a packet, in which case the exact position of a packet in the time interval is not predetermined.

Packets that are forwarded inside the network over the same route and in the same periodic time intervals constitute a virtual pipe and share the same pipe-ID. Pipe-ID can be either explicit, such as a tag or a label that is generated inside the network, or implicit such as a group of IP addresses. A virtual pipe can be used to transport data packets from multiple sources and to multiple destinations. A virtual pipe provides deterministic quality of service guarantees. The time interval in which a switch forwards a specific packet is determined by the packet's pipe-ID, the time it reaches the switch, and the current value of the common time reference. In accordance with the present invention, congestion-free packet switching is provided for pipe-IDs in which capacity in their corresponding forwarding links and time intervals is reserved in advance. Furthermore, packets that are transferred over a virtual pipe reach their destination in predefined time intervals, which guarantees that the delay jitter is smaller than or equal to one time interval.

Packets that are forwarded from one source to multiple destinations share the same pipe ID and the links and time intervals on which they are forwarded comprise a

virtual tree. This facilitates congestion-free forwarding from one input port to multiple output ports, and consequently, from one source to multiplicity of destinations. Packets that are destined to multiple destinations reach all of their destinations in predefined time intervals and with delay jitter that is no larger than one time interval.

A system is provided for managing data transfer of data packets from a source to a destination. The transfer of the data packets is provided during a predefined time interval, comprised of a plurality of predefined time frames. The system is further comprised of a plurality of switches. A virtual pipe is comprised of at least two of the switches interconnected via communication links in a path. A common time reference signal is coupled to each of the switches, and a time assignment controller assigns selected predefined time frames for transfer into and out from each of the respective switches responsive to the common time reference signal. For each switch, there is a first predefined time frame within which a respective data packet is transferred into the respective switch, and a second predefined time frame within which the respective data packet is forwarded out of the respective switch. The time assignment provides consistent fixed intervals between the time between the input to and output from the virtual pipe.

In a preferred embodiment, there is a predefined subset of the predefined time frames during which the data packets are transferred in the switch, and for each of the respective switches, there are a predefined subset of the predefined time frames during which the data packets are transferred out of the switch.

Each of the switches is comprised of one or a plurality of addressable input and output ports. A routing controller maps each of the data packets that arrives at each one of the input ports of the respective switch to a respective one or more of the output ports of the respective switch.

For each of the data packets, there is an associated time of arrival to a respective one of the input ports. The time of arrival is associated with a particular one of the predefined time frames. For each of the mappings by the routing controller, there is an associated mapping by a scheduling controller, which maps of each of the data packets between the time of arrival and forwarding time out. The forwarding time out is associated with a specified predefined time frame.

In the preferred embodiment, there are a plurality of the virtual pipes comprised of at least two of the switches interconnected via communication links in a path. The communication link is a connection between two adjacent switches; and each of the communications links can be used simultaneously by at least two of the virtual pipes. Multiple data packets can be transferred utilizing at least two of the virtual pipes.

In some configurations of this invention there is a fixed time difference, which is constant for all switches, between the time frames for the associated time of arrival and forwarding time out for each of the data packets. The fixed time difference is a variable time difference for some of the switches. A predefined interval is comprised of a fixed number of contiguous time frames comprising a time cycle. Data packets that are forwarded over a given virtual pipe are forwarded from an output port within a predefined subset of time frames in each time cycle. Furthermore, the number of data packets that can be forwarded in each of the predefined subset of time frames for a given virtual pipe is also predefined.

The time frames associated with a particular one of the switches within the virtual pipe are associated with the same switch for all the time cycles, and are also associated with one of input into or output from the particular respective switch.

In some configurations of this invention there is a constant fixed time between the input into and output from a respective one of the switches for each of the time frames within each of the time cycles. A fixed number of contiguous time cycles comprise a super cycle, which is periodic. Data packets that are forwarded over a given virtual pipe are forwarded from an output port within a predefined subset of time frames in each super cycle. Furthermore, the number of data packets that can be forwarded in each of the predefined subset of time frames within a super cycle for a given virtual pipe is also predefined.

In the preferred embodiment the common time reference signal is coupled from a GPS (Global Positioning System), and is in accordance with the UTC (Coordinated Universal Time) standard. The UTC time signal does not have to be received directly from GPS. Such signal can be received by using various means, as long as the delay or time uncertainty associated with that UTC time signal does not exceed half a time frame.

In one embodiment, the super cycle duration is equal to one second as measured using the UTC (Coordinated Universal Time) standard. The super cycle can also be equal to multiple UTC seconds or a fraction of a UTC second.

A select buffer controller maps one of the time frames for output from a first switch to a second time frame for input via the communications link to a second switch. The select buffer controller uses the UTC time signal in order to identify the boundaries between two successive time frames. The select buffer controller inserts a time frame delimiter (TFD) signal into the transmission link in order to the signal the second switch with the exact boundary between two time frames.

Each of the data packets is encoded as a stream of data, and a time frame delimiter is inserted into the stream of data responsive to the select buffer controller. This can be implemented by using a redundant serial codewords as it is later explained.

The communication links can be of fiber optic, copper, and wireless communication links for example, between a ground station and a satellite, and between two satellites orbiting the earth. The communication link between two nodes does not have to be a serial communication link. A parallel communication link can be used – such link can simultaneously carry multiple data bits, associated clock signal, and associated control signals.

The data packets can be Internet protocol (IP) data packets, and asynchronous transfer mode (ATM) cells, and can be forwarded over the same virtual pipe having an associated pipe identification (PID). The PID can be an Internet protocol (IP) address, Internet protocol group multicast address, an asynchronous transfer mode (ATM), a virtual circuit identifier (VCI), and a virtual path identifier (VPI), or (used in combination as VCI/VPI).

The routing controller determines two possible associations of an incoming data packet: (i) the output port, and (ii) the time of arrival (ToA). The ToA is then used by the scheduling controller for determining when a data packet should be forwarded by the select buffer controller to the next switch in the virtual pipe. The routing controller utilizes at least one of Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6) addresses, Internet protocol group multicast address, Internet MPLS (multi protocol label swapping or tag switching) labels, ATM virtual circuit identifier and virtual path identifier (VCI/VPI), and IEEE 802 MAC (media access control) addresses, for mapping from an input port to an output port.

Each of the data packets is comprised of a header, which includes an associated time stamp. For each of the mappings by the routing controller, there is an associated mapping by the scheduling controller, of each of the data packets between the respective associated time-stamp and an associated forwarding time out, which is associated with one of the predefined time frames. The time stamp can record the time in which a packet was created by its application.

In one embodiment the time-stamp is generated by an Internet real-time protocol (RTP), and by a predefined one of the switches. The time-stamp can be used by a scheduling controller in order to determine the forwarding time of a data packet from an output port.

Each of the data packets originates from an end station, and the time-stamp is generated at the respective end station for inclusion in the respective originated data packet. Such generation of a time-stamp can be derived from UTC either by receiving it directly from GPS or by using the Internet's Network Time Protocol (NTP).

1 Synchronous virtual pipe

In accordance with the present invention, a system is provided for transferring data packets across a data network while maintaining for reserved data traffic constant bounded jitter (or delay uncertainty) and no congestion-induced loss of data packets. Such properties are essential for many multimedia applications, such as, telephony and video teleconferencing.

In accordance with the design, method, and illustrated implementation of the present invention, one or a plurality of virtual pipes **25** are provided, as shown in FIGS. 1-2, over a data network with general topology. Such data network can span the globe. Each virtual pipe **25** is constructed over one or more switches **10**, shown in FIG. 1, which are interconnected via communication links **41** in a path.

FIG. 1 illustrates a virtual pipe **25** from the output port **40** of switch **A**, through switches **B** and **C**. This virtual pipe ends at the output port **40** of node **D**. The virtual pipe **25** transfers data packets from at least one source to at least one destination.

FIG. 2 illustrates three virtual pipes: virtual pipe 1 from the output of switch **A** to the output of switch **D**, virtual pipe 2 from the output of switch **B** to the output of switch **D**, and virtual pipe 3 from the output of switch **A** to the output of switch **C**.

The data packet transfers over the virtual pipe **25** via switches **10** are designed to occur during a plurality of predefined time intervals, wherein each of the predefined time intervals is comprised of a plurality of predefined time frames. The timely transfers of data packets are achieved by coupling a common time reference **002** (CTR) signal to each of the switches **10**.

FIG. 3 illustrates the structure of a pipeline switch **10**. The switch **10** is comprised of one or a plurality of input ports **30**, one or a plurality of output ports **40**, switching fabric **50**, and global positioning system (GPS) time receiver **20** with a GPS antenna **001**. The GPS time receiver provides a common time reference signal (CTR) **002** to all input and output ports.

1.1 The common time reference (CTR) 002

As shown in FIG. 4, the common time reference **002** that is coupled to the switches **10** provides the following property: the local clock ticks **004**, shown in FIG. 4, at all the pipeline switches (e.g., switches **A**, **B**, **C**, and **D** in FIGS. 1 and 2) when projected on the real-time axis **005** will all occur within predefined synchronization envelopes **003**. In other words, the local clock ticks **004** occur within the synchronization envelopes **003**, and therefore, outside relative to the synchronization envelopes all local clocks have the same clock value.

The common time reference is divided in a predefined manner into time frames, *Tf*, of equal duration, as shown in FIG. 4, typically *Tf* = 125 microseconds. The time

frames are grouped into time cycles. Each time cycle has predefined number of time frames.

Referring to FIG. 5, there are *k* time frames in each time cycle. Contiguous time cycles are grouped together into contiguous super cycles, and as shown in FIG. 5, there are *l* time cycles in each super cycle.

FIG. 6 illustrates how the common time reference can be aligned with the UTC (Coordinated Universal Time) standard. In this illustrated example, the duration of every super cycle is exactly one second as measured by the UTC standard. Moreover, the beginning of each super cycle coincides with the beginning of a UTC second, as shown in FIG. 6. Consequently, when leap seconds are inserted or deleted for UTC corrections (due to changes in the earth rotation period) the cycle and super cycle periodic scheduling will not be affected.

The time frames, time cycles, and super cycles are associated in the same manner with all respective switches within the virtual pipe at all times.

1.2 Pipeline forwarding

Pipeline forwarding relates to data packets being forwarded across a virtual pipe **25** with a predefined delay in every stage (either across a communication link **41** or across a switch **10** from input port **30** to output port **40**). Data packets enter a virtual pipe **25** from one or more sources and are forwarded to one or more destinations.

This sort of pipeline forwarding used in accordance with the present invention is illustrated in FIG. 7. Data packet **41A** is forwarded out of switch **A** during time frame *t-1*. This data packet **41A** will reach switch **B** after a delay of T-ab. This data packet **41A** will be forwarded out of switch **B** as data packet **41B** during time frame *t+1* and will reach switch **C** after a delay of T-bc. This data packet **41B** will be forwarded out of switch **C** as data packet **41C** during time frame *t+5*. Data packet **41C** will reach switch **D** after a delay of T-cd. Consequently, the delay from the output of switch **A** to the output of switch **C** is $6=t+5-(t-1)$ time frames. As illustrated in FIG. 7, all data packets that are forwarded over that virtual pipe will have a delay of six time frames from the output of switch **A** to the output of switch **C**.

Referring again to FIG. 1, the timely pipeline forwarding of data packets over the virtual pipe **25** is illustrated. A data packet is received by one of the input ports **30** of switch **A** at time frame 1, and is forwarded along this virtual pipe **25** in the following manner: (i) the data packet **41A** is forwarded from the output port **40** of switch **A** at time frame 2 of time cycle 1, (ii) the data packet **41B** is forwarded from the output port **40** of switch **B**, after 18 time frames, at time frame **10** of time cycle 2, (iii) the data packet **41C** is forwarded from the output port **40** of switch **C**, after 42 time frames, at time frame 2

of time cycle 7, and (iv) the data packet **41D** is forwarded from the output port **40** of switch **D**, after 19 time frames, at time frame 1 of time cycle 9.

As illustrated in FIG. 1,

- All data packets enter the virtual pipe **25** (i.e., forwarded out of the output port **40** of switch **A**) periodically at the second time frame of a time cycle, are output from this virtual pipe **25** (i.e., are forwarded out of the output port **40** of switch **D**) after 79 time frames.
- The data packets that enter the virtual pipe **25** (i.e., forwarded out of the output port **40** of switch **A**) can come from one or more sources and can reach switch **A** over one or more input links **41**.
- The data packets that exit the virtual pipe **25** (i.e., forwarded out of the output port **40** of switch **D**) can be forwarded over plurality of output links **41** to one of plurality of destinations.
- The data packets that exit the virtual pipe **25** (i.e., forwarded out of the output port **40** of switch **D**) can be forwarded simultaneously to multiple destinations, (i.e., multicast (one-to-many) data packet forwarding).
- The communication link **41** between two adjacent ones of the switches **10** can be used simultaneously by at least two of the virtual pipes.

In FIG. 2, where there are three virtual pipes:

- The three virtual pipes can multiplex (i.e., mix their traffic) over the same communication links.
- The three virtual pipes can multiplex (i.e., mix their traffic) during the same time frames and in an arbitrary manner.
- The same time frame can be used by multiple data packets from one or more virtual pipes.

1.3 Virtual pipe capacity assignment

For each virtual pipe there are predefined time frames within which respective data packets are transferred into its respective switches, and separate predefined time frames within which the respective data packets are transferred out of its respective switches. Though the time frames of each virtual pipe on each of its switches can be assigned in an arbitrary manner along the common time reference, it is convenient and practical to assign time frames in a periodic manner in time cycles and super cycles.

FIG. 8 illustrates the timing of a switch of a virtual pipe wherein there are a predefined subset of time frames (*i*, 75, and 80) of every time cycle, during which data packets are transferred into that switch, and wherein for that virtual pipe there are a predefined subset time frames (*i*+3, 1, and 3) of every time cycle, during which the data packets are transferred out of that switch. If each of the three data packets has 125 bytes

or 1000 bits, and there are 80 time frames of 125 microseconds in each time cycle (i.e., time cycle duration of 10msec), then the bandwidth allocated to this virtual pipe is 300,000 bits per second.

In general, the bandwidth or capacity allocated for a virtual pipe is computed by dividing the number of bits transferred during each of the time cycles by the time cycle duration. In the case of a super cycle, the bandwidth allocated to a virtual pipe is computed by dividing the number of bits transferred during each of the super cycles by the super cycle duration.

The switch **10** structure, as shown in FIG. 3, can also be referred to as a pipeline switch, since it enables a network comprised of such switches to operate as a large distributed pipeline architecture, as it is commonly found inside digital systems and computer architectures.

Each pipeline switch **10** is comprised of a plurality of addressable input ports **30** and output ports **40**. As illustrated in FIG. 12, the input port **30** is further comprised of a routing controller **35** for mapping each of the data packets that arrives at each one of the input ports to a respective one of the output ports. As illustrated in FIG. 16, the output port **40** is further comprised of a scheduling controller and transmit buffer **45**. An output port **40** is connected to an input port **30** via a communication link **41**, as shown in FIG. 9. The communication link can be realized using various technologies compatible with the present invention.

As shown in FIG. 3, the common time reference **002** is provided to the input ports **30** and output ports **40** from the GPS time receiver **20**, which receives its timing signal from the GPS antenna **001**. GPS time receivers are available from variety of manufacturers, such as, TrueTime, Inc. (Santa Rosa, CA). With such equipment, it is possible to maintain a local clock with accuracy of ±1 microsecond from the UTC (Coordinated Universal Time) standard everywhere around the globe.

1.4 The communication link and time frame delimiter encoding

The communication links **41** used for the system disclosed is in this invention can be of various types: fiber optic, wireless, etc. The wireless links can be between at least one of a ground station and a satellite, between two satellites orbiting the earth, or between two ground stations, as examples.

Referring to FIG. 9, a serial transmitter **49** and serial receiver **31** are illustrated as coupled to each link **41**. A variety of encoding schemes can be used for a serial line link **41** in the context of this invention, such as, SONET/SDH, 8B/10B Fiber Channel, 4B/5B FDDI (fiber distributed data interface). In addition to the encoding and decoding of the data transmitted over the serial link, the serial transmitter/receiver (**49** in FIG. 12 and **31** in FIG. 16) sends/receives control words for a variety of control purposes,

mostly unrelated to the present invention description. However, one control word, time frame delimiter (TFD), is used in accordance with the present invention. The TFD marks the boundary between two successive time frames and is sent by a serial transmitter **49** when a CTR **002** clock tick occurs in a way that is described hereafter as part of the output port operation. It is necessary to distinguish in an unambiguous manner between the data words, which carry the information, and the control signal or words (e.g., the TFD is a control signal) over the serial link **41**. There are many ways to do this. One way is to use the known 4B/5B encoding scheme (used in FDDI). In this scheme, every 8-bit character is divided into two 4-bit parts and then each part is encoded into a 5-bit codeword that is transmitted over the serial link **41**.

FIG. 10 illustrates an encoding table from 4-bit data to 5-bit serial codewords. The 4B/5B is a redundant encoding scheme, which means that there are more codewords than data words. Consequently, some of the unused or redundant serial codewords can be used to convey control information.

FIG. 11 is a table with 15 possible encoded control codewords, which can be used for transferring the time frame delimiter (TFD) over the serial link. The TFD transfer is completely transparent to the data transfer, and therefore, it can be sent in the middle of the data packet transmission in a non-destructive manner.

When the communication links **41** are SONET/SDH, the time frame delimiter cannot be embedded as redundant serial codewords, since SONET/SDH serial encoding is based on scrambling with no redundancy. Consequently, the TFD is implemented using the SONET/SDH frame control fields: transport overhead (TOH) and path overhead (POH). Note that although SONET/SDH uses a 125 microseconds frame, it cannot be used directly in accordance with the present invention, at the moment, since SONET/SDH frames are not globally aligned and are also not aligned to UTC. However, if SONET/SDH frames are globally aligned, SONET/SDH can be used compatibly with the present invention.

1.5 The input port

As shown in FIG. 12, the input port **30** has three parts: serial receiver **31**, a routing controller **35** and separate queues to the output ports **36**. The serial receiver **31** transfers the data packets and the time frame delimiters to the routing controller **35**.

The routing controller **35** is constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data packet, read only memory (ROM) for storing the routing controller processing program and the routing table that is used for determining the output port that the incoming data packet should be switched to.

As illustrated in FIG. 13, the incoming data packet header includes a virtual pipe identification, PID **35C**, that is used to lookup in the routing table **35D** the address **35E**

of the queue **36** that the incoming data packet should be transferred into. Before the packet is transferred into its queue **36**, the time of arrival (ToA) **35T** is attached to the packet header as illustrated in FIGS 15A and 15B. The ToA **35T** is used by the scheduling controller **45** of the output port **40** in the computation of the forwarding time out of the output port, and shown in FIG. 16.

The data packet can have various formats, such as, Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6), asynchronous transfer mode (ATM) cells, etc. The data packets PID can be determined by one of the following: an Internet protocol (IP) address, an asynchronous transfer mode (ATM) a virtual circuit identifier, a virtual path identifier (VCI/VPI), Internet protocol version 6 (IPv6) addresses, Internet MPLS (multi protocol label swapping or tag switching) labels, and IEEE 802 MAC (media access control) address, etc..

FIG. 14 illustrates the flow chart for the router controller **35** processing program executed by the routing controller **35B**. The program is responsive to two basic events from the serial receiver **31** of FIG. 12: the received time frame delimiter TFD at step **35-01**, and the receive data packet at step **35-02**. After receiving a TFD, the routing controller **35** computes the time of arrival (ToA) **35T** value at step **35-03** that is attached to the incoming data packets. For this computation it uses a constant, **Dconst**, which is the time difference between the common time reference (CTR) **002** tick and the reception of the TFD at time *t2* (generated on an adjacent switch by the CTR **002** on that node). This time difference is caused by the fact that the delay from the serial transmitter **49** to the serial receiver **31** is not an integer number of time frames. When the data packet is received at step **35-02**, the routing controller **35B** executes three operations as set forth in step **35-04**: attach the ToA, lookup the address of the queue **36** using the PID, and storing the data packet in that queue **36**.

1.6 The switching fabric

There are various ways to implement a switching fabric. However, the switching fabric is peripheral to the present invention, and so it will be described only briefly. The main property that the switching fabric should ensure is that packets for which the priority bit P (**35P** in FIGS. 15A and 15B) is set to high, then priority (i.e., reserved traffic) will be switched into the output port in a constant bounded delay - measured in time frames.

This is possible in accordance with the present invention, where the packets in the input ports are already separated into queues to their respective output ports. Then, by using the Clos theorem in the time domain (see J.Y. Hui, "Switching and Traffic Theory for Integrated Broadband Networks", page 65), the delay can be bounded by two time frames, one time frame at the input port and one time frame to get across the

switching fabric. Other implementations can be used, such as based on shared bus with round robin service of the high priority data packets, or on a crossbar switch.

Another possible switch design is shared memory, which ensures a deterministic delay bound from an input port to an output port. Shared memory packet switches are commercially available from various vendors, for example, MMC Networks Inc. (Santa Clara, CA).

FIGS. 15A and 15B illustrate data packets without and with a time stamp attached, respectively.

1.7 The output port

The output port **40** is illustrated in FIG. 16, comprised of a scheduling controller with a transmit buffer **45**, and serial transmitter **49** (as previously described herein). The scheduling controller **45** performs a mapping of each of the data packets between the associated respective time of arrival (ToA) and an associated forwarding time out of the output port via the serial transmitter **49**. The forwarding time is determined relative to the common time reference (CTR) **002**.

Three output port configurations are illustrated herein: a double-buffer scheduling controller, as shown in FIGS. 17 and 18, a general scheduling controller, as shown in FIGS. 19, 20, and 21, and a general scheduling controller with time-stamp, as shown in FIGS. 22 and 23.

The double-buffer scheduling controller **46**, as illustrated in the block diagram of FIG. 17 and flow chart of FIG. 18, is constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data packet, and read only memory (ROM) for storing the controller processing program. Each time frame as specified by the common time reference **002** is considered to be one of an even tick or an odd tick. The determination of even tick vs. odd tick is made relative to the beginning of a time cycle. In the preferred embodiment, the first time frame of a time cycle is determined to be an odd tick, the second time frame of the time cycle is determined to be an even tick, the third time frame of the time cycle is determined to be an odd tick, and so forth, where the determination of even tick vs. odd tick alternates as shown for the duration of the time cycle. In an alternate embodiment, the first time frame of a time cycle is determined to be an even tick, the second time frame of the time cycle is determined to be an odd tick, the third time frame of the time cycle is determined to be an even tick, and so forth, where the determination of even tick vs. odd tick alternates as shown for the duration of the time cycle. The actual sequence of even ticks vs. odd ticks of time frames within a time cycle may be arbitrarily started with no loss in generality.

The double-buffer scheduling controller **46** operates in the following manner. Data packets arrive from the switching fabric **50** via link **51**. When the priority bit **35P**

is asserted (i.e., reserved traffic), the packet is switched through the packet DMUX (demultiplexer) **51S** (during odd ticks of the common time reference) to buffer **Ba** via link **51-1**, and during even ticks of the common time reference to buffer **Bb**, via link **51-2**. Data packets in which the priority bit **35P** is not asserted (i.e., non-reserved traffic) are switched through the packet DMUX (demultiplexer) **51S** to the "best effort" buffer **Bc** via link **51-3**. The transmit buffer selection operation is controlled by the select signal **46A**, which connects the double-buffer scheduling controller with the packet DMUX (demultiplexer) **51S**.

Data packets are forwarded to the serial transmitter **49** through the packet MUX (multiplexer) **47S**, and link **47C** in FIG. 17, during odd ticks of the common time reference from buffer **Bb** via link **46-2**, and during even ticks of the common time reference from buffer **Ba** via link **46-1**. If during odd ticks of the common time reference buffer **Bb** is empty, data packets from the "best effort" buffer **Bc** are forwarded to the serial transmitter. If during even ticks of the common time reference buffer **Ba** is empty, data packets from the "best effort" buffer **Bc** are forwarded to the serial transmitter. The transmit buffer selection operation is controlled by the select signal **46B**, which connects the double-buffer scheduling controller **46** with the packet MUX (multiplexer) **47S**.

A more general scheduling controller **45** operation is described in FIGS. 19, 20, and 21, which includes a transmit buffer **45C** and a select buffer controller **45D**. The data packet scheduling controller **45A**, together with the select buffer controller **45D**, perform the mapping, using the PID **35C** and the data packet time of arrival (ToA) **35T** in order to determine the respective time frame a respective packet should be forwarded out of the output port. Both controllers **45A** and **45D** are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

Data packets arrive from the switching fabric **50** via link **51**. Data packets which have the priority bit **35P** asserted (i.e., reserved traffic) are switched by the scheduling controller **45A** to one of the k transmit buffers **45C** (B-1, B-2, ..., B-*k*). Each of the *k* buffers is designated to store packets that will be forwarded in each of the *k* time frames in every time cycle, as shown in FIG. 5.

The flow chart for the program executed by the scheduling controller is illustrated in FIG. 20. When the data packet is received from the fabric at step **45-03**, the PID **35C** in the data packet header is used to look-up the forward parameter **45F** in the forwarding table (**45B** of FIG. 19), as specified in step **45-04**. Next, the index *i* of the transmit buffer, between B-1 and B-*k'*, is computed in step **45-05** by subtracting the time of arrival ToA **35T** from the common time reference CTR **002** and by adding the

forward parameter **45F**, and then switching the incoming data packet to transmit buffer B-*i*, as specified in step **45-06**.

Incoming data packets in which the priority bit **35P** is not asserted (i.e., non-reserved traffic) are switched by the scheduling controller to the transmit "best effort" buffer B-E via link **45-be**.

FIG. 21 illustrates the flow chart for the select buffer controller **45D** operation. The controller **45D** is responsive to the common time reference (CTR) tick **002**, and at step 45-11, increments the transmit buffer index *i* (i.e., *i:=i+1 mod k'*, where *k'* is the number of buffers for scheduled traffic) and sends a time frame delimiter TFD to the serial transmitter at step **45-12**. Then, if the transmit buffer B-*i* is not empty, at step **45-13**, it will send a data packet from transmit buffer B-*i*, as specified in at step **45-14**, else it will send a "best effort" data packet from the "best effort" buffer **B-be**, as specified at step **45-15**.

FIGS. 22 and 23 illustrate a system with a scheduling controller, wherein each of the data packets is comprised of a header, including an associated time stamp. The time-stamp is generated by an Internet real-time protocol (RTP) in which its data packet format is illustrated in FIG. 22. Alternatively, the time-stamp can be generated by a predefined one of the switches **10** in the system, or the time stamp can be generated at a respective end station for inclusion in the respective originated data packet.

FIG. 23 illustrates the operation of the scheduling controller for the case where the packet header contains a time-stamp **35TS**. Data packets arrive from the switching fabric **50** via link **51**. Data packets in which the priority bit **35P** is set (i.e., reserved traffic) are switched by the scheduling controller to one of the *k* transmit buffers **45C** (B-1, B-2,..., B-*k*). Each of the *k* buffers is designated to store packets that will be forwarded in each of the *k* time frames in every time cycle, as shown in FIG. 5. The flow chart for the program executed by the scheduling controller is illustrated in FIG. 23. When a data packet is received from the fabric at step **45-21**, the PID **35C** in the data packet header is used to look-up the forward parameter **45F** in the forwarding table **45B**, as specified in step **45-22**. Next the index *i* of the transmit buffer, between B-1 and B-*k*, is computed in step **45-23** by subtracting the time of arrival ToA **35T** from the common time reference CTR **002** and by adding the forward parameter **45F**, and then switching the incoming data packet to transmit buffer B-*i*, as specified in step **45-24**.

2 Time-based routing

In this variant of this invention, a packet that arrives to an input port of a switch is switched to an output port based on (i) its position within the predefined time interval and (ii) the unique address of the incoming input port. Each switch along a route from a source to a destination forwards packets in periodic time intervals that are predefined

using the common time reference. The time interval duration can be longer than the time duration required for transmitting a packet.

FIG. 24 depicts a schematic description of a switch **10**. The switch **10** is constructed of four components: a plurality of uniquely addressable input ports **30** (in FIG. 24 there are N such ports), a plurality of uniquely addressable output ports **40** (in FIG. 24 there are N such ports), a switching fabric **50**, and a global positioning system (GPS) time receiver **20** with a GPS antenna **001**. The GPS time receiver provides a common time reference (CTR) **002** to all input and output ports. The common time reference is partitioned into time frames. Each of the time frames is further comprised of predefined positions such that the input port can unambiguously identify the positions. The time and position that a data packet arrives into the input port are used by the routing controller **35** in FIG. 12 for determining the output port that incoming data packet should be switched to.

In FIG. 24, each of the time frames, $t=i$ and $t=i+1$, has four predefined positions: a, b, c and d. In each of the positions, one data packet can be stored. The positions can be marked explicitly with position delimiters (PDs) between the variable size data packets, as it will be explained below, or implicitly. Implicit position within a time frame can be achieved by either measuring time delays - this is suitable for sending a fixed size ATM (asynchronous transfer mode) cells, or by placing data packets of variable size in the predefined positions within each of the time frames – if the output port **40** does not have a data packet to transmit in a predefine position an empty or null data packet should be sent.

FIG. 25 depicts a common time reference (CTR) **002** axis that is divided into time cycles. Each time cycle is divided into predefined frames. Each of the time frame has predefined positions: a, b, c, and d of either fixed size (in time duration) or variable size (in time duration), consequently, the predefined position can have ether fixed size data packets or variable size data packets, respectively.

2.1 The time-based input port

The input port **30**, shown in FIG. 12, has three parts: serial receiver **31**, time-based routing controller **35** and separate queues **36** to the plurality of output ports **40**. The serial receiver **31** transfers to the time-based routing controller **35** data packets, time frame delimiters (TFD) and position delimiters (PD).

The routing controller is constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data packets, read only memory (ROM) for storing the time-based routing controller processing program, and a time-based routing table is used for determining the following parameters (see **35D** in FIGS. 26, 27, and 29):

1. Parameter **35-1** in table **35D** (FIGS. 27 and 29) – the output port **40** that the incoming data packet should be switched to – this parameter is used for switching the data packet to the queue **36** that is leading to the corresponding output port;

2. Parameter **35-2** in table **35D** (FIGS. 27 and 29) – the out-going time frame in which the data packet will be forwarded out of the output port – this parameter is attached to the data packet header in FIG. 27, and

3. Parameter **35-3** in table **35D** (FIGS. 27 and 29) – the position within the out-going time frame in which the data packet will be forwarded out of the output port – this parameter is attached to the data packet header in FIG. 28.

The time-based routing controller **35B** determines the entry to the time-based routing table **35D**, in FIGS. 27 and 29, in various ways, such as:

1. Local time and position by using (1) the time frame of arrival (ToA) **35T** – the time frame using the common time reference **002**, and (2) the position value **35P** within that time frame as measured by the position counter **35PC**. This is depicted in FIG. 12.

2. Time stamp **35TS** and position **35PC** by using (1) the time stamp **35TS** in the data packet header in FIG. 28B, and (2) the position value **35P** within that time frame as measured by the position counter **35PC**.

3. Time stamp, PID (shown in the packet headers in FIG. 28) and position **35PC** by (1) the time stamp **35TS** in the data packet header in FIG. 28B, (2) the virtual pipe ID (PID) **35C** in the data packet header in FIG. 28B (the virtual pipe is discussed in details at the end of this description), and (3) the position value **35P** within that time frame as measured by the position counter **35PC**. This is depicted in FIG. 29.

The data packets, see for example FIG. 28, can have various formats, such as, Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6), asynchronous transfer mode (ATM) cells. The data packets PID **35C** can be determined by one of the following: an Internet protocol (IP) address, an asynchronous transfer mode (ATM), a virtual circuit identifier, and a virtual path identifier (VCI/VPI), Internet protocol version 6 (IPv6) addresses, Internet MPLS (multi protocol label swapping or tag switching) labels, and IEEE 802 MAC (media access control) address.

The time stamp **35TS** in the packet header in FIG. 28B can be generated by an application using Internet real-time protocol (RTP) and is used also in the ITU-T H.323 standard. Such data packets use the format depicted in FIG. 22. Alternatively the time-stamp can be generated by a predefined one of the switches in the system, or alternatively the time stamp is generated at the respective end node for inclusion in the respective originated data packet.

FIG. 30 is a detailed description of the program executed by the time-based routing controller **35B**. The program is responsive to three events from the serial receiver **31** and the position value **35P** within that time frame as measured by the position counter **35PC**. The time-based routing controller program FIG. 30 using the three parameters in table **35D** in FIGS. 27 and 29 that is associated with this incoming packet operates as follows:

1. Receive time frame delimiter TFD **35-1** – responsive to this event the routing controller resets the position counter (**35P**:=0 in **135-04** of FIG. 30) and computes the time-frame of arrival (ToA) **35T** value as specified in **135-04** of FIG. 30. For this computation it uses a constant, **Dconst**, which is the time difference between the common time reference (CTR) **002** tick and the reception of the TFD at time *t2* (note that the TFD was generated on an adjacent switch by the CTR **002** on that node). This time difference is caused by the fact that the delay from the serial transmitter **49** to the serial receiver **31** is not an integer number of time frames.

2. Receive position delimiter PD **135-02** – responsive to this event it increments the position counter, **35P**:=**35P**+1, **135-05** of FIG. 30.

3. Receive data packet **135-03** –responsive to this event three operations are performed as shown in **135-06** of FIG. 30: (1) the out-going time frame parameter **35-2** is attached to the packet header, (2) the position within the out-going time frame parameter **35-3** is attached to the packet header, and (3) the data packet is stored in the queue **36** using the output port parameter **35-1** in table **35D** in FIGS. 27 and 29.

2.2 The time-based output port

The output is depicted in FIG. 16, it has two parts a scheduling controller with a transmit buffer **45**, and serial transmitter **49**, which was described before. The data packet scheduling controller **45A**, in FIG. 19, transfers the data packet the transmit buffer which is a random access memory (RAM) **45C**, as described below.

The data packet scheduling controller **45** operation is described in FIGS. 19, 31, and 32, which includes a transmit buffer **45C** and a select buffer controller **45D**. The scheduling controller **45A** together with the select buffer controller **45D** perform the mapping, using the two parameters, **35-2** and **35-3**, that were attached to the data packet by the routing controller **35B**. Both controllers are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

Data packets that arrive from the switching fabric **50** via link **51** in which their priority bit **35P** is asserted (i.e., reserved traffic) will be switched by the data packet scheduling controller **45A** to one of the *k'* transmit buffers **45C**: B-1, B-2, ..., B-*k'* (one special case is when *k'*=*k*, where *k* is the time cycle size measured in time frames). Each

of the *k'* buffers is designated to store packet that will be forwarded in a cyclically recurring order in each of the *k* time frames in every time cycle, as shown in FIGS. 5 and 6. The actual program executed by the data packet scheduling controller is described in FIG. 31. When data packet is received from the fabric **145-01** (in FIG. 31) the two parameters, **35-2** and **35-3**, in the data packet header are used to determine in which of the transmit buffer, between B-1 and B-*k'*, to store that data packet and in what position, as specified in **145-02** in FIG. 31.

Incoming data packets in which their priority bit **35P**, see FIG. 28, is not asserted (i.e., non-reserved traffic) will be switched by the data packet scheduling controller to the transmit "best effort" buffer B-E via link **45-be**.

FIG. 32 depicts the select buffer controller **45D** operation, which is responsive to the common time reference (CTR) tick **002**, as specified in **145-11** (FIG. 32). Consequently, the select buffer controller increments the transmit buffer index *i* **145-12** (i.e., $i:=i+1 \bmod k'$, where *k'* is the number of buffers for scheduled traffic), sends a time frame delimiter TFD **47A** to the serial transmitter **145-12**, and reset the position pointer to one, p:=1 **145-12**. Then while the transmit buffer B- *i* is not empty **145-13**, it will send a data packets from transmit buffer B- *i*, as specified in **145-14**, **145-15** and **145-16**, else if the transmit buffer B- *i* is empty, it will send "best effort" data packets from the "best effort" buffer **B-be**, as specified in **145-17**, until the end of the time frame (the next CTR **002** tick) or until buffer **B-E** becomes empty.

When the transmit buffer **B-** *i* is not empty **145-13**, the select buffer controller sends data packets from all of the non-empty predefined positions in that buffer, as specified in **145-14**. After sending a data packet or if position p in buffer B- *i* is empty **145-15**, the select buffer controller sends a position delimiter (PD) **47B** to the serial transmitter and increments the position pointer p:=p+1, as specified in **145-16**.

3 Traffic integration

3.1 The integrated input port

The input port **30**, shown in FIG. 12, has three parts: serial receiver **31**, routing controller **35** and separate queues to the output ports **36**. The serial receiver **31** transfers to the routing controller **35** the data packets and the time frame delimiters.

The routing controller is constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data packet, read only memory (ROM) for storing the routing controller processing program, and routing table is used for determining the output port that the incoming data packet should be switched to. The incoming data packet header includes a virtual pipe identification - PID **35C** in FIG. 34, that is used to lookup in the routing table **35D** the address **35E** of the queue the incoming data packet should be transferred into its queue **36**. Before the packet is

transferred into its queue **36** the time of arrival (ToA) **35T** in FIG. 34 is attached to the packet header. The ToA **35T** will be used by the scheduling controller **45** in FIG. 16 in the computation of the forwarding time out of the output port.

The data packet can have various formats, such as, Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6), asynchronous transfer mode (ATM) cells. The data packets PID can be determined by one of the following: an Internet protocol (IP) address, an asynchronous transfer mode (ATM) a virtual circuit identifier, and a virtual path identifier (VCI/VPI), Internet protocol version 6 (IPv6) addresses, Internet MPLS (multi protocol label swapping or tag switching) labels, and IEEE 802 MAC (media access control) address.

FIG. 35 is a table for defining two bits, P1 and P2, in the packet headers in FIG. 34. The two bits classify three types of data packets: P1/P2 are "00" constant bit rate (CBR) data traffic; P1, P2 are 01 variable bit rate (VBR) data traffic; and P1, P2 are "10" "best effort" data traffic. The above classification is used by the program executed by the routing controller **35B**, as shown in FIG. 36, in order to determine into which of the three parts of the queue to the output port **36**, shown in FIG. 33, the data packet should be switched into.

FIG. 36 is a detailed description of the program executed by the routing controller **35B**. The program is responsive to two basic events from the serial receiver **31**: receive time frame delimiter TFD **235-01**, and receive data packet **235-02**. After receiving a TFD the routing controller computes the time of arrival (ToA) **35T** value **235-03** in FIG. 34, that is attached to the incoming data packets. For this computation it uses a constant, Dconst, which is the time difference between the common time reference (CTR) **002** tick and the reception of the TFD at time *t2* (note that the TFD was generated on an adjacent switch by the CTR **002** on that node). This time difference is caused by the fact that the delay from the serial transmitter **49** to the serial receiver **31** is not an integer number of time frames. When data packet is received **235-02** the routing controller **35B** executes three operations **235-04** in FIG. 36: attach the ToA, lookup the address of the queue **36** using the PID, and storing the data packet in the queue **36** to the output port **37**, while using P1/P2 in the header, in FIG. 34, in order to determine in what part, CBR/VBR/Best effort, of that queue to store the incoming data packet.

3.2 Integrated scheduling controller in the output port

A more general scheduling controller **45** operation is described in FIGS. 37-39, which includes a scheduling and rescheduling controller **145A**, a transmit buffer **145C**, and a select buffer and congestion controller **145D**, as shown in FIG. 37. The scheduling and rescheduling controller **145A** together with the select buffer controller **145D** perform the mapping of the data packet into the time frame. The mapping is done

on the scheduling and rescheduling controller using the PID **35C** and the data packet time of arrival (ToA) **35T** in order to determine the respective time frame in which the respective packet should be forwarded out of the output port. The details are presented below. Both controllers, **145A** and **145D**, are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

In the more general configuration, depicted in FIG. 37, data packets that arrive from the switching fabric **50** via link **51** in which their priority bits **35P** (P1/P2) are either "00" or "01" (i.e., reserved CBR traffic, or VBR traffic, respectively) will be switched by controller **145A** to one of the k transmit buffers in **145C**: B-1, B-2, ..., B-*k*. Each of the *k* buffers is designated to store packet that will be forwarded in each of the *k* time frames in every time cycle, that where defined in FIGS. 5 and 6. Another possible operation is to map the incoming packets separately to each of the time frames of a super-cycle. When a super-cycle mapping is implemented there are *k*l* transmit buffers in **145C**: B-1, B-2, ..., B-*k*l*, i.e., *k* buffers to each of the *l* cycles of a super-cycle.

The actual program executed by the scheduling and rescheduling controller is described in FIG. 38. When a data packet is received from either the fabric via link **51** or from the select buffer and congestion controller **145D** via link **45R**, as specified in **145-03**, the **35C**, **35T** and **35P** in the data packet header are used to look-up the forward parameter **45F** in the forwarding table **145B**, as specified in **145-04**. Next, the index *i* of the transmit buffer, between B-1 and B-*k*, is computed in **145-05** by subtracting the time of arrival ToA **35T** from the common time reference CTR **002** and by adding the forward parameter **45F**, and then switching the incoming data packet to transmit buffer B-*i*, as specified in **145-05**.

Incoming data packets in which their priority bits **35P**, P1/P2, are either "10" (i.e., non-reserved traffic or "best effort") or "11" (i.e., rescheduled packet) are switched by the scheduling and rescheduling controller to the transmit "best effort" buffer B-E via link **45-be**.

FIG. 39 depicts the operation of the select buffer and congestion controller **145D** operation, which is responsive to the common time reference (CTR) tick **002**. When CTR signal is received **245-11** the following operations are executed by controller **45D** in **245-15**:

1. Send time frame delimiter (TFD) control signal **47A** to the serial transmitter **49**;

2. Forward back to controller **145A** all unsent packets in transmit buffer B-f in which the **35P** field in their header is not "11", i.e., it is not a rescheduled packet and

set the 35P field to "11", i.e., a rescheduled packet (note that a packet can be rescheduled only once);

3. If the number of data packets need to be rescheduled exceeds some predefined number, say *n*, then select at random *n* data packets and sends them back to the packet scheduling and rescheduling controller **145A** via link **45R** and discard the remainder of the packets.

4. Increment the transmit buffer index f (i.e., $f:=f+1 \bmod k'$, where k' is the number of buffers for scheduled traffic). If the CBR part of buffer B-f is not empty **245-12**, then it will send a data packet from transmit buffer B-f first CBR packets and then VBR packets, as specified in **245-16** and **245-13**, else it will send a "best effort" data packet from the "best effort" buffer B-E as specified in **245-14**.

4 Operation with links with variable delay

The present invention further relates to a system and method for transmitting and forwarding packets over a packet switching network in which some of its communication links have dynamically varying delays. Such variations in the link delay can be the consequence of having mobile switching node (e.g., satellites).

FIG. 40 illustrates a virtual pipe **25** from the output port **40** of switch **A**, through switches **B** and **C**. This virtual pipe ends at the output port **40** of node **D**. The virtual pipe **25** transfers data packets from at least one source to at least one destination. In FIG. 40 the communication link that connects switch **B** to switch **C** may have a delay that varies in time with a defined delay bound. Such communication links are found in various network architectures, such as, mobile wireless networks, satellite networks, and self-healing SONET rings. In satellite networks the delay between a satellite in space and a base-station on earth changes in the following manner. First the delay decreases, as the satellite appears above the horizon and is moving towards the base-station, and then the delay increases as the satellite is moving away from the base-station until it disappears below the horizon.

FIGS. 41 and 42 describe the delay changes on the link between Nodes **B** and **C** as it is projected on a common time reference (CTR), which is discussed in details in FIGS. 4, 5, and 6 above. FIG. 41 describes a communication link between Nodes **B** and **C**, where the time of arrival to Node **C** decreases, i.e., the delay between Nodes **B** and **C** gets shorter. In FIG. 41, the delay of data packet **0a** is longer than data packet **1b**, the delay of data packet **1b** is longer than data packet **2c**, and so on. FIG. 42 describes a communication link between Nodes **B** and **C**, where the time of arrival to Node **C** increases, i.e., the delay between Nodes **B** and **C** gets longer. In FIG. 42, the delay of data packet **0a** is shorter than data packet **1b**, the delay of data packet **1b** is shorter than data packet **2c**, and so on. A complete description of the above resynchronization

operation is part of the output operation described below in FIGS. 46, 47, 48, 50, and 51.

FIGS. 43, 44, and 45 describe a delay variations that are due to the forwarding of successive data packets on alternate paths or routes in the network. FIG. 43 shows two virtual pipes (defined below), *p* (from Node **A** to **B** to **C** to **D** and to **E**) and *p'* (from Node **A** to **B'** to **D'** and to **E**), with the requirement that data packets will be forwarded out of Node **E** at the same predefined time regardless which virtual pipe, *p* or *p'*, they were forwarded on. FIG. 44 shows the scenario in which the data packets on virtual pipe *p'* arrive to Node **E** before the time they would have arrived to Node **E** on virtual pipe *p*. Consequently, the data packets on path *p'* should be delayed, as shown in FIG. 44 and forwarded, in time, as if they have arrived on virtual pipe *p*. FIG. 45 shows how such resynchronization can be achieved by using a resynchronization buffer on node **E**. In FIG. 45, a data packet from virtual pipe *p'*, 1 *p'*, enters a resynchronization buffer **10R**, such that, when this data packet exits this buffer, 2 *p'*, it will be forwarded from the output of Node **E** as if this packet was forwarded on virtual pipe *p*. A complete description of the above resynchronization operation is part of the output port operation is described below in FIGS. 16, 46, 47, 48, 49, 50, and 51.

The output port **40** is illustrated in FIG. 16, comprised of a scheduling controller with a transmit buffer **45**, and serial transmitter **49** (as previously described herein). The scheduling controller **45** performs a mapping of each of the data packets between the associated respective time of arrival (ToA) and an associated forwarding time out of the output port via the serial transmitter **49**. The forwarding time is determined relative to the common time reference (CTR) **002**.

The scheduling controller and transmit buffer **45** has various modes of operation which are described in FIGS. 46, 47, 48, 49, 50, and 51. The different operation modes correspond to some of the possible variations in the communications link delay as was discussed in FIGS. 40-45. The scheduling controller and transmit buffer **45** in FIG. 46 includes three parts:

1. a delay analysis and scheduling controller **245A** which further comprises a forwarding table **245B**,

2. a transmit buffer **245C** which is typically realized as a random access memory (RAM), and

3. a select buffer controller **245D** which forward data packets to the serial transmitter.

The delay analysis and scheduling controller **245A**, together with the select buffer controller **245D**, perform the mapping, using the PID **35C**, the time-stamp **35TS** and the data packet time of arrival (ToA) **35T** in order to determine the respective time

frame a respective packet should be forwarded out of the output port. Both controllers **245A** and **245D** are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

Data packets arrive from the switching fabric **50** via link **51**. Data packets which have the priority bit **35P** asserted (i.e., reserved traffic) are switched by the delay analysis and scheduling controller **245A** to one of the $l*k$ transmit buffers **45C** (B-1, B-2..., B-$l*k$). Each of the $l*k$ buffers is designated to store packets that will be forwarded in each of the $l*k$ time frames in every super cycle, as shown in FIG. 5 and FIG. 6. Having $l*k$ transmit buffers enables the delay analysis and scheduling controller **245A** to schedule data packets in a wide range of delay variations. When the super cycle is one second the scheduling capability of the delay analysis and scheduling controller **245A** is up to one second. However, this is an extreme case and in most practical scenarios the scheduling requirements, even with delay varying links, is only a small number of time frames.

The transmit buffer **245C** includes an additional buffer **B-E** for "best effort" data packets. The priority bit **35P** in the "best effort" data packets is not asserted and this how the delay analysis and scheduling controller determines that such data packets should be stored in the "best effort" buffer. The "best effort" data packets are forwarded to the serial transmitter **49** whenever there are no more scheduled data packets (priority bit **35P** asserted).

FIG. 47 illustrates the flow chart for the select buffer controller **45D** operation. The controller **45D** is responsive to the common time reference (CTR) tick **002** at step **345-11**, and then, at step **345-12**, it increments the transmit buffer index i (i.e., $i:=i+1 \ mod \ l'*k'$, where $l'*k'$ is the number of buffers for scheduled traffic), and sends a time frame delimiter TFD to the serial transmitter **49**. Then, if the transmit buffer B-i is not empty, at step **345-13**, it will send a data packet from transmit buffer B-i, as specified in at step **345-14**, else it will send a "best effort" data packet from the "best effort" buffer B-E, as specified at step **345-15**.

The flow chart for the program executed by the delay analysis and scheduling controller is illustrated in FIG. 48. The main task of the program is to compute the index, i, of the transmit buffer, B-i, between B-1 and B-$l'*k'$, is computed in step **245-05**. There are several possible methods to perform the computation in step **245-05**, which depends on the type of delay variations that can occur on the communication links. In FIGS. 49 and 50 three possible computation methods are described:

I. FIG. 49 - the case of continuous delay variations as described in FIGS. 40-42, as specified in **45-051**:

1. Let <s1, s2, s3, ... , *sj*> be the set of time frames of a PID=*p*, which repeats in every super cycle, as it is specified in the forwarding table **245B** at the *p* entry,
2. Controller **245A** searches the set <s1, s2, s3, ... , *sj*> in order to determine the first feasible time frame, *si*, that occur after (ToA **35T**)+CONST (where CONST is a constant bound on the delay across the switching fabric, and
3. *si* is the time frame the data packet is scheduled for transmission via the serial transmitter - where i is the index transmit buffer B-*i*.

The set <s1, s2, s3, ... , *sj*> constitute plurality of time frame in which a data packet can be scheduled for transmission out of the output port of a switch.

II. FIG. 50 - the case of multiple path with resynchronization as described in FIGS. 43, 44, and 45: When the data packet is received from the fabric at step **245-03**, the PID **35C** in the data packet header is used to look-up the resynchronization parameter **45R** in the forwarding table (**245B** of FIG. 46, as specified in step **45-151**, and then in step **45-152**: Compute the index, *i*, of the transmit buffer **245C**: *i* = [(ToA **35T**)+**45R**] mod *l'*k'* (where *l'*k'* is the number of buffers for scheduled traffic). In the case of two virtual pipes: *p* and *p'*, where one is an alternative to the other, as shown in FIGS. 43, 44, and 45, the above resynchronization is needed on both ends. More specifically resynchronization is needed on both Node **A** and Node **E**, as shown in FIGS. 43, 44, and 45.

III. FIG. 51 - the case when using a time-stamp in the packet header, FIGS. 15 and 22: When the data packet is received from the fabric at step **45-03**, the PID **35C** in the data packet header is used to look-up the forward parameter **45F** in the forwarding table (**245B** of FIG. 46), as specified in step **45-251**, and then in step **45-252**: Compute the index, *i*, of the transmit buffer **245C**: *i* = [(Time-stamp **35TS**)+**45F**] mod *l'*k'* (where *l'*k'* is number of buffers for scheduled traffic).

5 Monitoring, policing and billing

The present invention further relates to a system and method for monitoring, policing and billing of the transmission and forwarding of data packets over a packet switching network. The switches of the network maintain a common time reference, which is obtained either from an external source (such as GPS - Global Positioning System) or is generated and distributed internally.

The output port **40** is illustrated in FIG. 16, comprised of a scheduling controller with a transmit buffer **45**, serial transmitter **49** (as previously described herein), and the monitoring and policing controllers. The scheduling controller **45** performs a mapping of each of the data packets between the associated respective time of arrival (ToA) and

an associated forwarding time out of the output port via the serial transmitter **49**. The forwarding time is determined relative to the common time reference (CTR) **002**.

A general scheduling controller **45** operation was previously described in FIGS. 19-21, which includes a transmit buffer **45C** and a select buffer controller **45D**. The data packet scheduling controller **45A**, together with the select buffer controller **45D**, perform the mapping, using the PID **35C** and the data packet time of arrival (ToA) **35T** in order to determine the respective time frame a respective packet should be forwarded out of the output port. Both controllers **45A** and **45D** are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

5.1 The monitoring and policing controllers

The monitoring and policing controllers **65** (FIGS. 52-55) are part of both the input port in FIG. 12 and the output port in FIG. 16. Monitoring and policing controllers **65** are of two basic types:

1. The delay monitoring controller **65D** - for ensuring the correct timing behavior by PID=*p* (FIGS. 52, 53).
2. The policing and load controller **65P** - for ensuring the correct capacity usage by PID=*p* (FIGS. 54, 55).

Both controllers **65D** and **65P** are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

5.2 The delay monitoring controller 65D

FIGS. 52 and 53 describe the operation of a delay monitoring controller **65D**. This controller checks data packets in which their reserved priority bit, **35P** in their headers, is asserted for three cases:

1. Data packet is within two predefined delay parameters range (see box **65D-05**): between the two delay parameters: **65-par-L** and **65-par-H**, which were found PID=*p* **35C** entry in the parameters table **65-Tab** (see box **65D-02**). More specifically, the delay monitoring controller **65D** computes the actual delay the data packet already experienced: **65-Del** = Time of arrival **35T** - Time-stamp **35TS** (see box **65D-03**), then comparing that it is in the predefined delay range: (**65-Del** > **65 - Par-L** and **65-Del** < **65 - Par-H**) (see box **65D-04**).
2. Data packet is late (see box **65D-07**): its delay is greater than **65-par-H**, i.e., **65-Del** >**65 - Par-H** (see box **65D-06**), and
3. Data packet is early (see box **65D-08**): its delay is smaller than **65-par-L**, i.e., **65-Del** < **65 - Par-L**.

The three cases have importance on ensuring proper network operations and the adherence to the user quality of service (QoS) requirements. Furthermore, the information collected by the delay monitoring controller is reported to upper layer protocols, which are outside the scope of this invention.

5.3 The policing and load controller **65P**

FIGS. 54 and 55 describe the operation of an alternate embodiment of policing and load controller **65P** that checks and ensures that a data packets in which its reserved priority bit **35P** (in its header) is asserted will not exceed the predefined load of its virtual pipe PID=p. This controller operation can be used for both:

1. Policing - detecting PID=p that exceeds its reserved capacity, and
2. Billing - recording the actual capacity usage of PID=p.

The two cases have importance on ensuring proper network operations and the adherence to the user quality of service (QoS) requirements.

When a data packet is received (see box **65P-01**) the policing an load controller **65P** first computes the current load, L(p) for PID=p (see box **65P-02**) by L(p): = L(p)+1 (see box **65P-02**) using the load table **65L** that stores previous values of L(p). The load can be computed in various ways: (i) per time frame of PID=p, (ii) per time cycle of PID=p, or (iii) per super cycle of PID=p.

Next the policing and load controller **65P** using the PID=p **35C** looks-up the parameter **65-Par** in the table **65-Tab**. Then if **65-Par**>L(p) (see box **65P-03**) the data packet is dropped (see box **65P-05**), otherwise the data packet is forwarded (see box **65P-04**).

In the above two cases, the load L(p) information on PID=p is recorded and reported to upper layer protocols for billing the usage for the usage of PID=p. Furthermore, the policing and load information is used also for ensuring proper network operations and the adherence to the user quality of service (QoS) requirements. The information collected by the policing and load controller is reported to upper layer protocols, which are outside the scope of this invention.

6 Interconnecting a synchronous with an asynchronous switching networks

The present invention further relates to a system and method for transmitting and forwarding packets over a heterogeneous packet switching network in which some of its some of its switches are synchronous and some switches are asynchronous. The invention specifically ensures that the synchronous switches will forward data packets in predefined time intervals although are arriving to the synchronous switches with relatively large, but bounded, delay uncertainty. Such delay uncertainty is the consequence of having asynchronous switches on the route of a data packet before it reaches the synchronous switch.

A system is provided for managing in a timely manner transfer of data packets across asynchronous switches with three configurations:

1. From an end-station **100** across asynchronous LAN switches **20** to a synchronous virtual pipe switch **10**, as shown in FIG. 56. In this embodiment, an end-station **100** sends data packets from its network interface **102**. The data packet is switched through one or more asynchronous switches **20** (e.g., Nodes **B** and **C** in FIG. 56) until it reaches the synchronous virtual pipe switch **10**. In this configuration, the synchronous switch **10** will resynchronized the incoming data packet as it is explained below.
2. From an end-station **100** network interface **102** across asynchronous LAN switches **20** to an asynchronous to synchronous gateway **15**, which converts the asynchronous data packets stream to a synchronous stream and then forward the data packets to a synchronous virtual pipe switch, as shown in FIG. 57.
3. From a first synchronous virtual pipe switch **10** across asynchronous switches or routers **20** to a second synchronous virtual pipe switch **10**, as shown in FIG. 58. In this configuration, the incoming data packet to the second synchronous virtual pipe switch **10** will be resynchronized as it is explained below.

6.1 Resynchronization by the router controller at the input port

A data packet that has been forwarded by one or asynchronous switches **20**, i.e., switches without common time references can be resynchronized to its original schedule at either the input port or the output port. FIGS. 59-61 describe the resynchronization operation at the input port of either a synchronous virtual pipe switch **10** or a gateway **15**.

In order to facilitates the resynchronization of data packets sent by the end-station **100**, the synchronous virtual pipe switch **10** periodically sends timing messages **M002** to the end-station **100**, as shown in FIG. 59. The timing messages **M002** represent the value of the common time reference (CTR) **002** as it is received by the synchronous switch **10**.

The network interface **102** at the end-station **100** incorporates the timing information obtained from the timing message **M002** into the time-stamp field **35TS** (FIG. 15) in the header of the data packets, **1DP**, **2DP**, **3DP**, ... , **9DP** in FIG. 59, it sends across the asynchronous switches **20** to the synchronous virtual pipe switch **10**. FIG. 60 show the possible time of arrival of data packets, **1DP**, **2DP**, **3DP**, ... , **9DP**, to the input port **35**. A data packet can experience delay that is ranging between minimum

delay and maximum delay, which constitute the delay bound uncertainty to be Maximum delay - Minimum delay, as shown in FIG. 60.

FIGS. 61A and 61B show the resynchronization operation performed by the routing controller **35** at the input port **30** using a resynchronization buffer **30R**, which is executed in the following two steps:

1. Using the PID **35C** look-up the delay bound uncertainty parameter **35DB** (which is [Maximum delay - Minimum delay]) in the routing table **35B**.
2. Delay the packet at the input port by: **35DB** - [(CTR **002**) - (Time-stamp **35TS**)].

6.2 The router controller operation with a gateway

When there is an asynchronous to synchronous gateway **15** before the input to virtual pipe switch the router controller operates as was previously specified herein in FIG. 14. In this case, the gateway receives the CTR **002** and also forward time frame delimiter (TFD) **47A** to the serial receiver, as it will be specified below. More specifically, its operation can be identical to the output operation which will be described below.

FIG. 14 illustrates the flow chart for the router controller **35** processing program executed by the routing controller **35B**. The program is responsive to two basic events from the serial receiver **31** of FIG. 14: the receive time frame delimiter TFD at step **35-01**, and the receive data packet at step **35-02**. After receiving a TFD, the routing controller **35** computes the time of arrival (ToA) **35T** value at step **35-03** that is attached to the incoming data packets. For this computation it uses a constant, **Dconst**, which is the time difference between the common time reference (CTR) **002** tick and the reception of the TFD at time t2 (generated on an adjacent switch by the CTR **002** on that node). This time difference is caused by the fact that the delay from the serial transmitter **49** to the serial receiver **31** is not an integer number of time frames. When the data packet is received at step **35-02**, the routing controller **35B** executes three operations as set forth in step **35-04**: attach the ToA, lookup the address of the queue **36** using the PID, and storing the data packet in that queue **36**.

6.3 The router controller operation without a gateway

When the virtual pipe switch **10** is connected directly to the asynchronous network the virtual pipe switch **10** cannot receive TFD, and therefore, the operation of the routing controller, as specified was previously specified herein in FIG. 14, should be changed in the following way. The operations in **35-01** and **35-03** are taken out and in part one of the operation specified in **35-04**, in FIG. 14, should clearly state that the time of arrival value attached to the packet header should be derived directly from the common time reference (CTR) **002**.

6.4 The output port

The output operation described herein applies to both the synchronous virtual pipe switch **10**, in FIGS. 56 and 58, and the asynchronous to synchronous gateway **15**, in FIG. 57.

The output port **40** was previously specified herein in FIG. 16, comprised of a scheduling controller with a transmit buffer **45**, and serial transmitter **49** (as previously described herein). The scheduling controller **45** performs a mapping of each of the data packets between the associated respective time of arrival (ToA) and an associated forwarding time out of the output port via the serial transmitter **49**. The forwarding time is determined relative to the common time reference (CTR) **002**. As it will be described, this mapping resynchronized the stream of data packets forwarded by the virtual pipe switch.

The scheduling controller and transmit buffer **45** has various modes of operation which are described in FIGS. 19, 48–49, and 62. The different operation modes corresponds to some of the possible configurations with the asynchronous switches, as was discussed in FIGS. 56-58. The scheduling controller and transmit buffer **45** in FIG. 19 includes three parts:

1. a delay analysis and scheduling controller which further comprises a forwarding table **45B**;
2. a transmit buffer **45C** which is typically realized as a random access memory (RAM); and
3. a select buffer controller **45D** which forward data packets to the serial transmitter.

The delay analysis and scheduling controller **45A**, together with the select buffer controller **45D**, perform the mapping, using the PID **35C**, the time-stamp **35TS** and the data packet time of arrival (ToA) **35T** in order to determine the respective time frame a respective packet should be forwarded out of the output port. Both controllers **45A** and **45D** are constructed of a central processing unit (CPU), a random access memory (RAM) for storing the data, and read only memory (ROM) for storing the controller processing program.

Data packets arrive from the switching fabric **50** via link **51**. Data packets which have the priority bit **35P** asserted (i.e., reserved traffic) are switched by the delay analysis and scheduling controller **45A** to one of the k transmit buffers **45C** (B-1, B-2, ..., B-k). Each of the k buffers is designated to store packets that will be forwarded in each of the k time frames in every time cycle, as shown in FIGS. 5 and 6. Having k transmit buffers enables the delay analysis and scheduling controller **45A** to schedule data packets in a wide range of delay variations.

The transmit buffer **45C** includes an additional buffer B-E for "best effort" data packets. The priority bit **35P** in the "best effort" data packets is not asserted and this is how the delay analysis and scheduling controller determine that such data packets should be stored in the "best effort" buffer. The "best effort" data packets are forwarded to the serial transmitter **49** whenever there are no more scheduled data packets (priority bit **35P** asserted).

6.5 The select buffer controller operation:

FIG. 21 illustrates the flow chart for the select buffer controller **45D** operation. The controller **45D** is responsive to the common time reference (CTR) tick **002** at step **45-11**, and then, at step **45-12**, it increments the transmit buffer index *i* (i.e., $i:=i+1 \bmod k'$, where k' is the number of buffers for scheduled traffic) and sends a time frame delimiter TFD to the serial transmitter **49**. Then, if the transmit buffer B-*i* is not empty, at step **45-13**, it will send a data packet from transmit buffer B-*i*, as specified in at step **45-14**, else it will send a "best effort" data packet from the "best effort" buffer B-E, as specified at step **45-15**.

6.6 The delay analysis and scheduling controller

The flow chart for the program executed by the delay analysis and scheduling controller is illustrated in FIG. 48. The main task of the program is to compute the index, *i*, of the transmit buffer, B-*i*, between B-1 and B-*k*, is computed in step **245-05**. There are several possible methods to perform the computation in step **245-05**, which depends on the type of delay variations that can occur on the communication links. In FIGS. 49 and 62, two possible computation methods are described:

I. FIG. 49 - the case of arbitrary, but bounded, delay variations as described in FIGS. 56-58, as specified in **45-051**:

1. Let <s1, s2, s3, ... , s*j*> be the set of time frames of a PID=p, which repeats in every time cycle, as it is specified in the forwarding table **45B** at the p entry,
2. controller **45A** searches the set <s1, s2, s3, ... , s*j*> in order to determine the first feasible time frame, s*i*, that occur after (ToA 35T)+CONST (where CONST is a constant bound on the delay across the switching fabric); and
3. s*i* is the time frame the data packet is scheduled for transmission via the serial transmitter - where *i* is the index transmit buffer B-*i*.

The set <s1, s2, s3, ... , s*j*> constitute plurality of time frame in which a data packet can be scheduled for transmission out of the output port of a switch.

II. FIG. 62 - the case when using a time-stamp in the packet header (FIGS. 15 and 22): When the data packet is received from the fabric at step **45-03**, the PID 35C in

the data packet header is used to look-up the forward parameter 45F in the forwarding table (45B of FIG. 19), as specified in step 45-351, and then in step 45-352: Compute the index, i, of the transmit buffer 45C: i = [(Time-stamp 35TS)+45F] mod k', where k' is the number of buffers for scheduled traffic.

From the foregoing, it will be observed that numerous variations and modifications may be effected without departing from the spirit and scope of the invention. It is to be understood that no limitation with respect to the specific apparatus illustrated herein is intended or should be inferred. It is, of course, intended to cover by the appended claims all such modifications as fall within the scope of the claims.

WHAT IS CLAIMED IS:

1. A system for scheduling data packets on a switch, said system comprising:

a common time reference signal coupled to the switch;

wherein the common time reference is partitioned into time frames;

a plurality of input ports each having a unique address;

a plurality of output ports each having a unique address;

a memory partitioned into plurality of buffers;

a select buffer controller for selecting one of the plurality of buffers.

a switching fabric for coupling incoming data packets between selected ones of the input ports and the output ports;

wherein the transfer of the data packets is provided during respective ones of a plurality of predefined time intervals;

wherein each of the predefined time intervals is comprised of a plurality of predefined time frames; and

wherein one or more packets can be communicated within each of the time frames.

2. The system as in claim 1, further comprising:

a routing controller with a routing table for selecting at least one output port that said data packets will be forwarded to;

wherein the routing controller attaches a time of arrival (ToA) to incoming data packets at the input port;

wherein the time of arrival (ToA) relates to the common time reference and is represented as a time frame number within a time cycle and as a time cycle number within a super cycle.

3. The system as in claim 2, wherein a predefined number of contiguous k time frames are grouped into a time cycle; wherein k is at least 1;

wherein a predefined number of contiguous l time cycles are grouped into a super cycle; and wherein l is at least 1.

4. The system as in claim 2, further comprising:

a data packet scheduling controller responsive to the reception of a data packet from the switching fabric and assigns a selected predefined time frame for transferring said data packet out from the said switch responsive to at least one of the time of arrival attached to the respective data packet by the routing controller, the unique identity of the respective input port, and the PID in the data packet header.

5. The system as in claim 2, wherein when there are no scheduled data packets to be transmitted in a time frame, "best effort" data packets are transmitted.

6. The system as in claim 2, wherein a time frame delimiter (TFD) is realized by at least one of the following methods: by transmitting a TFD signal between every two successive time frames and by determining it directly from the common time reference at the receiving node.

7. The system as in claim 6, wherein the time frame delimiter can be transmitted as an encoded control codeword in the middle of a data packet without adversely affecting the transmission of said data packet.

8. The system as in claim 6, wherein all data packets received between two successive time frame delimiters (TFDs), which were received by the routing controller, are uniquely associated with the time frame as measured by the common time reference.

9. The system as in claim 2, wherein the memory is partitioned into three buffers;

wherein a first buffer stores data packets to be forwarded from said switch during odd time frames as measured by the common time reference;

wherein a second buffer stores data packets to be forwarded from said switch during even time frames as measured by the common time reference; and

wherein a third buffer stores "best effort" data packets to be forwarded from said switch during one of the odd and even time frames, whenever the respective first and second buffers are empty.

10. The system as in claim 3, wherein the memory is partitioned into $k'+1$ buffers;

wherein each of a first k' buffers provides a scheduled buffer for storing data packets to be forwarded from said switch during one of the k time frames in each of said time cycle, as were measured by the common time reference; and

wherein a $k'+1$ buffer provides for storing "best effort" data packets.

11. The system as in claim 10, wherein k' is equal to k such that in each of the k time frames of the time cycle, the select buffer controller forwards data packets from the corresponding one of the k buffers; and

wherein when any of said other k' buffers are empty, said select buffer controller forwards out of said switch "best effort" data packets from the $k'+1$ buffer.

12. The system as in claim 10,

wherein k' is not equal to k and the select buffer controller forwards data packets from the k' buffers in a cyclically recurring order such that in every time frame, data packets are forwarded from a different one of the k' buffers;

wherein when said scheduled buffer is empty, said select buffer controller forwards out of said switch "best effort" data packets from the $k'+1$ buffer.

13. The system as in claim 3, wherein the memory is partitioned into [(l' times k')+1] buffers;

wherein a first (l' times k') buffer provides for storing data packets to be forwarded from said switch during one of the (l' times k') time frames in each of said super cycle, as were measured by the common time reference;

wherein the [(l' times k')+1] buffer provides for storing "best effort" data packets.

14. The system as in claim 13, wherein (l' times k') is equal to (l times k) such that in each of the (l times k) time frames of the super cycle, the select buffer controller forwards data packets from the respective corresponding one of the (l' times k') buffers;

wherein when the respective corresponding one of said buffers is empty, said select buffer controller forwards out of said switch "best effort" data packets from said [l'(times k') +1] buffer.

15. The switching system as in claim 1, further comprising:

a timing subsystem for determining a defined time in as an associated time-frame of arrival (ToA) for each of the incoming data packets, for associating each said respective data packets with a particular one of the predefined time frames;

position logic for determining a relative position for each of said respective incoming data packet within the respective particular time frame; and

a time-based routing controller responsive to (1) the unique address of the input port associated with each one of the incoming data packets; (2) associated time-frame of arrival (ToA); and (3) associated relative position for each said respective incoming data packet within said time-frame of arrival, to provide a routing to an associated particular one of the output ports at an associated particular position and within an associated second particular time frame.

16. The system as in claim 15, wherein the time-based routing controller provides routing of each of the incoming data packets to a plurality of the output ports, each of the plurality of the incoming data packets having a respective unique associated particular position within an associated one of the predefined time frames.

17. The system as in claim 15, wherein each of the frames has a defined duration from start to end, wherein each of the input ports is comprised of a serial receiver that provides position delimiters associated with each of the incoming data packets coupled thereto, wherein the position logic is a position counter, wherein said position counter counts the position delimiters that have occurred since the start of the respective one of the time frames.

18. The system as in claim 17, wherein the time frames are cyclically recurring, wherein a sequence of position delimiters within the cyclically recurring time frames is implicitly defined by a predefined sequence of time units of equal duration.

19. The system as in claim 17, wherein the time frames are cyclically recurring, wherein the sequence of position delimiters within the cyclically recurring time frames is defined by a predefined sequence of time intervals of arbitrarily different duration.

20. The system as in claim 17, wherein the start of the time frame is marked by a time frame delimiter.

21. The system as in claim 17, wherein the position logic is incremented at predefined time intervals relative to the start of the time frame.

22. The system as in claim 21, wherein each of the positions is of predefined time duration.

23. The system as in claim 15, wherein the time interval is comprised of a predefined number of contiguous time frames comprising a time cycle.

24. The system as in claim 23, wherein the time frames in time cycle are of equal time duration.

25. The system as in claim 23, wherein the time frames in the time cycle are of arbitrary time duration.

26. The system as in claim 23, wherein the time cycles are contiguous.

27. The system as in claim 23, wherein each of the time cycles consists of an arbitrary number of time frames.

28. The system as in claim 23, wherein the switching system is comprised of a plurality of switches interconnected via communication links, wherein the time frames associated with a particular one of the switches are associated with the same respective switch for all the time cycles.

29. The system as in claim 28, wherein the time frames associated with the particular ones of the switches are associated with one of input into and output from that respective switch.

30. The system as in claim 29, wherein there is a constant predefined fixed delay measured in time frames between the input into the input port and output from the output port of the same switch for each of the time frames within each of the time cycles.

31. The system as in claim 29, wherein there is an arbitrary predefined delay measured in time frames between the input into the input port and output from the output port of the switch for each of the time frames within each of the time cycles.

32. The system as in claim 26, wherein a fixed number of a plurality of the contiguous ones of the time cycles comprise a super cycle;

wherein the super cycle is periodic.

33. The system as in claim 1, wherein the common time reference signal is derived from a GPS (Global Positioning System) signal.

34. The system as in claim 2, wherein the common time reference signal conforms to the UTC (Coordinated Universal Time) standard.

35 The system as in claim 34, wherein the super cycle duration is equal to at least one of the following: one second as measured using the UTC standard, a predefined number of seconds as measured using the UTC standard, and a predefined fraction of a second measured using the UTC standard.

36. The system as in claim 15, wherein a fixed plurality of the time frames comprises a time cycle, wherein the time frames have positions within the time cycles, wherein the position of the time frame within the time cycle and the position of the incoming data packets within the time frame determines the routing to a respective one of the output ports independent of which one of the time cycles the time frame is in.

37. The system as in claim 15, wherein a fixed plurality of the time frames comprise a time cycle, wherein a fixed number of a plurality of the time cycles comprises a super cycle, wherein routing is determined uniquely within a super cycle and is the same for all the time cycles within all the super cycles.

38. The system as in claim 37, wherein the routing is determined responsive to the respective time cycle within the super cycle associated an input to a respective one of the input ports, and the position of the respective data packet within each of the time frame.

39. The system as in claim 15, wherein there are a plurality of synchronization envelopes, each defining acceptable plus and minus deviations associated with the common time reference;

wherein two adjacent ones of the synchronization envelopes of two of the adjacent time frames are non-overlapping;

wherein each of the synchronization envelopes has one of the common time references uniquely associated with the respective synchronization envelope.

40. The system as in claim 1, further comprising:

means for switching data packets from at least one source to at least one destination, wherein the transfer of the data packets is provided during respective ones of a plurality of predefined time frames with predefined time intervals, wherein each of the predefined time intervals is comprised of a plurality of the predefined time frames, and within a predefined position within the time frame,

a time-based routing controller at the input port for determining uniquely to which of the output ports a data packet should be switched responsive to the time-frame of arrival (ToA) of said data packet at the input port, and position of said data packet within said time-frame of arrival (ToA);

a scheduling controller and transmit buffer for forwarding data packets out of the respective output port at a predefined scheduled time frame and position within said time frame.

41. The system as in claim 40, wherein a predefined number of contiguous ones of the time frames are grouped into a time cycle;

wherein a predefined number of contiguous time cycles are grouped into a super cycle; and

wherein the common time reference reflects the UTC time and is represented as a time frame number within a time cycle, and as a time cycle number within the super cycle.

42. The system as in claim 40, wherein a position delimiter signal is output between every two successive time frames responsive to the common time reference;

wherein the time-based routing controller is further comprised of:

a time-based controller;

a position counter for counting the position delimiter signals in the time frames associated with incoming packets, and for providing an instant position counter value; and

a time-based routing table that determines (i) the output ports said respective data packet will be switched to, (ii) a time frame associated with output of the respective data packet, and (iii) position within a time frame for the respective data packet to be output from the output port, responsive to (i) the time of arrival (ToA) and (ii) the instant position counter value.

43. The system as in claim 42, wherein the data packets are comprised of a packet header having a time stamp;

wherein the time of arrival is determined responsive to the time stamp.

44. The system as in claim 42, wherein the data packets are comprised of a packet header, and wherein the time frame associated with output and the position within the time frame for the respective output port are attached to become a part of the packet header.

45. The system as in claim 43, wherein the scheduling controller is responsive to the time of arrival (ToA) and within said time of arrival to provide an associated out-going time frame and the position in said out-going time frame for output from the output port.

46. The system as in claim 45, wherein the scheduling controller stores data packets in a random access memory buffer responsive to the out-going time frame and the position within said out-going time frame.

47. The system as in claim 3, further comprising:

a position counter for providing an instant position counter value responsive to counting the position delimiter signals in incoming time frames associated with the incoming data packets;

a routing controller with a time-based routing table for selecting at least one output port that said data packets will be forwarded to;

wherein the routing controller is coupled to the input port and attaches to the packet header a time-frame of arrival (ToA) and the instant position counter value prior to coupling of the respective data packets to the output port;

wherein the time-frame of arrival is representative of the common time reference and is represented as a time frame number within a time cycle and as a time cycle number within a super cycle; and

an output data packet scheduling controller for scheduling predefined time frames associated with prospective output of each of the incoming data packets.

48. The system as in claim 47, wherein the data packet scheduling controller is responsive to the reception of a data packet from the switching fabric for selectively assigning a predefined time frame and a position within said time frame for transferring said respective data packet out from the said switch responsive to (i) the time-frame of arrival, and (ii) the instant position counter value that were attached to the data packet by the time-based routing controller, and (iii) the unique address of the input port.

49. The system as in claim 47, further comprising a random access memory partitioned into plurality of buffers;

wherein the random access memory is partitioned into $k'+1$ buffers;

wherein first k' buffers are for storing data packets to be forwarded from said switch during respective associated ones of the k time frames in each of said time cycles;

wherein the $k'+1$ buffer provides for storing "best effort" data packets not associated uniquely with any predefined time frame.

50. The system as in claim 49, further comprising a select buffer controller;

wherein in each of the k time frames of the time cycle, the select buffer controller forwards the respective data packets from the corresponding one of the k' buffers;

wherein when said corresponding one k' buffer is empty, said select buffer controller forwards out of said $k'+1$ buffer the "best effort" data packets.

51. The system as in claim 50, wherein between every two successive data packets the select buffer controller sends a position delimiter (PD) signal via the serial transmitter.

52. The system as in claim 3, further comprising:

a routing controller with a routing table for selecting at least one of the output ports said data packets will be forwarded to;

wherein the routing controller is coupled to the input port and provides for attaching a time of arrival (ToA) to incoming ones of the data packets;

wherein the time of arrival is derived from the common time reference and is represented as a time frame number within a time cycle and as a time cycle number within a super cycle;

a packet scheduling and rescheduling controller for scheduling respective ones of the data packets for transfer during predefined associated ones of the time frames.

53. The system as in claim 52, further comprising a select buffer and congestion controller;

wherein the packet scheduling and rescheduling controller is responsive to the reception of a data packet for assigning a first predefined time frame for transferring said data packet out from the said switch when capacity is available responsive to the time of arrival attached to the data packet by the routing controller, the unique address of input port, and the PID in the data packet header;

wherein the select buffer and congestion controller determines if capacity is available in said first predefined time frame for transferring said data packet out from the said switch; and

wherein the packet scheduling and rescheduling controller is responsive to the reception of the data packet from the select buffer and congestion controller for assigning a second predefined time frame for transferring said data packet out from the said switch when the capacity is not available.

54. The system as in claim 53, further comprising a random access memory partitioned into plurality of buffers;

wherein the random access memory is partitioned into three buffers;

wherein the first buffer stores data packets to be forwarded from said switch during odd time frames as measured by the common time reference;

wherein the second buffer stores data packets to be forwarded from said switch during even time frames as measured by the common time reference;

wherein the third buffer stores "best effort" data packets to be forwarded from said switch during odd and even time frames, whenever the first and second buffers are empty.

55. The system as in claim 54, wherein said first buffer and said second buffer are further partitioned into two parts (i) constant bit rate (CBR) and (ii) variable bit rate (VBR).

56. The system as in claim 55, wherein during each time frame the select buffer and congestion controller first transfers data packets from the CBR part and then from the VBR part of said buffer.

57. The system as in claim 52, further comprising random access memory, wherein the random access memory is partitioned into $k'+1$ buffers;

wherein first k' buffers provide for storing data packets to be forwarded from said switch during one of the k time frames in each of said time cycle;

wherein the $k'+1$ buffer provides for storing "best effort" data packets; and

wherein said first k' buffers are further partitioned into two parts (i) constant bit rate (CBR) and (ii) variable bit rate (VBR).

58. The system as in claim 57, further comprising a select buffer and congestion controller,

wherein in each of the k time frames of the time cycle, the select buffer and congestion controller forwards data packets from the corresponding one of the k' buffers;

wherein said select buffer and congestion controller forwards data packets first from the CBR part of said buffer and then from the VBR part of said buffer, and when said VBR part is empty, said select buffer and congestion controller forwards data packets out of said "best effort" buffer.

59. The system as in claim 52, further comprising a select buffer and congestion controller and a random access memory,

wherein the random access memory is partitioned into (l times k)+1 buffers;

wherein first (l times k) buffers provide for storing data packets to be forwarded from said switch during one of the (l times k) time frames in each of said super cycle;

wherein the (l times k)+1 buffer provides for storing "best effort" data packets; and

wherein said first (l times k) buffers are further partitioned into two parts (i) constant bit rate (CBR) and (ii) variable bit rate (VBR).

60. The system as in claim 59, wherein in each of the (l times k) time frames of the super cycle, the select buffer and congestion controller forwards the respective data packets from the corresponding one of the (l times k) buffers;

wherein said select buffer and congestion controller forwards the respective data packets first from the CBR part of said buffer, and then from the VBR part of said buffer, and wherein the select buffer and congestion controller forwards data packets out of said "best effort" buffer when the said VBR part is empty.

61. The system as in claim 52, further comprising a select buffer and congestion controller, wherein between every two successive time frames the select buffer and congestion controller sends a time frame delimiter (TFD) signal via a serial transmitter.

62. The system as in claim 61, wherein the time frame delimiter can be sent as an encoded control codeword in the respective data packet without adversely affecting the transmission of said data packet.

63. The system as in claim 3, further comprising:

a data packet header having a PID field;

apparatus for determining availability of transmission capacity for each of the time frames;

a delay analysis and scheduling controller for assigning a first feasible time frame, from a selected plurality of predefined time frames, for transfer of the respective data packet out from the respective switch responsive to the time of arrival, the unique address of the respective input port, and the PID field in the data packet header for the respective data packet, and the availability of transmission capacity in said time frame;

a random access memory partitioned into plurality of buffers for storing the respective data packets in associated ones of the buffers associated with the respective time frames; and

a select buffer controller for selecting one of the plurality of buffers for output.

64. The system as in claim 63, wherein the data packet scheduling controller is responsive to the reception of a respective one of the data packets from the switching fabric for assigning a first feasible time frame from a predefined plurality of selected predefined time frames for transferring said respective data packet out from the said switch responsive to the time of arrival attached to the data packet by the routing controller, the unique address of the respective input port, and the PID in the respective data packet header.

65. The system as in claim 64, wherein the first feasible time frame is selected as the first time frame with available transmission capacity for forwarding said data packet.

66. The system as in claim 63, wherein the random access memory is partitioned into $k'+1$ buffers;

wherein first k' buffers are for storing data packets to be forwarded from said switch during one of the k time frames in each of said time cycle, as were measured by the common time reference;

wherein the $k'+1$ buffer is for storing "best effort" data packets.

67. The system as in claim 63, wherein in each of the k time frames of the time cycle, the select buffer controller provides for forwarding of the data packets from one of the k' buffers in a predefined cyclically recurring order;

wherein when said first k' buffer is empty, said select buffer controller forwards said "best effort" data packets from the $k'+1$ buffer out of said switch.

68. The system as in claim 3, further comprising:

a system for transferring data packets from at least one end-station to a data communications network;

interconnected communication links in a path in the network for connecting the end-station to said switch through said network;

wherein there is a defined delay for transfer of the respective data packets between the end-station and the switch; and

a delay analysis and scheduling controller, for determining for the switch, a first scheduled time within a first predefined time frame within which a respective one of the data packets is scheduled to be transferred out of the respective switch.

69. The system as in claim 68, wherein the connection of the end-station to the said switch is asynchronous.

70. The system as in claim 68, wherein the delay analysis and scheduling controller computes the delay in time frames that the data packet is to be delayed before being forwarded out of the respective output port of the switch.

71. The system as in claim 68, wherein a time-stamp value is attached to each said data packet when it is forwarded by the end-station;

wherein the time-stamp value is derived from the common time reference;

wherein the delay analysis and scheduling controller computes the time difference in time frames that the data packet is to be delayed before being forwarded out of the respective output port of the switch, responsive to the time-stamp value.

72. The system as in claim 71, wherein the common time reference is transferred to the end-station by a timing message carried in a data packet from the switch.

73. The system as in claim 71, wherein the common time reference is coupled directly to the end-station.

74. The system as in claim 71, wherein the common time reference is transferred to the end-station by a timing message carried in a data packet from a timing server.

75. The system as in claim 68, wherein the delay and analysis scheduling controller is responsive to the reception of a respective one of the data packets received from the switching fabric, and assigns a first feasible time frame from a plurality of predefined time frames, for transferring said respective data packet out from the respective switch, responsive to the respective time of arrival, and the unique address of the input port of the switch.

76. The system as in claim 75, wherein each of the data packets has a PID field, wherein the delay analysis and scheduling controller is responsive to the respective PID field.

FIG. 1

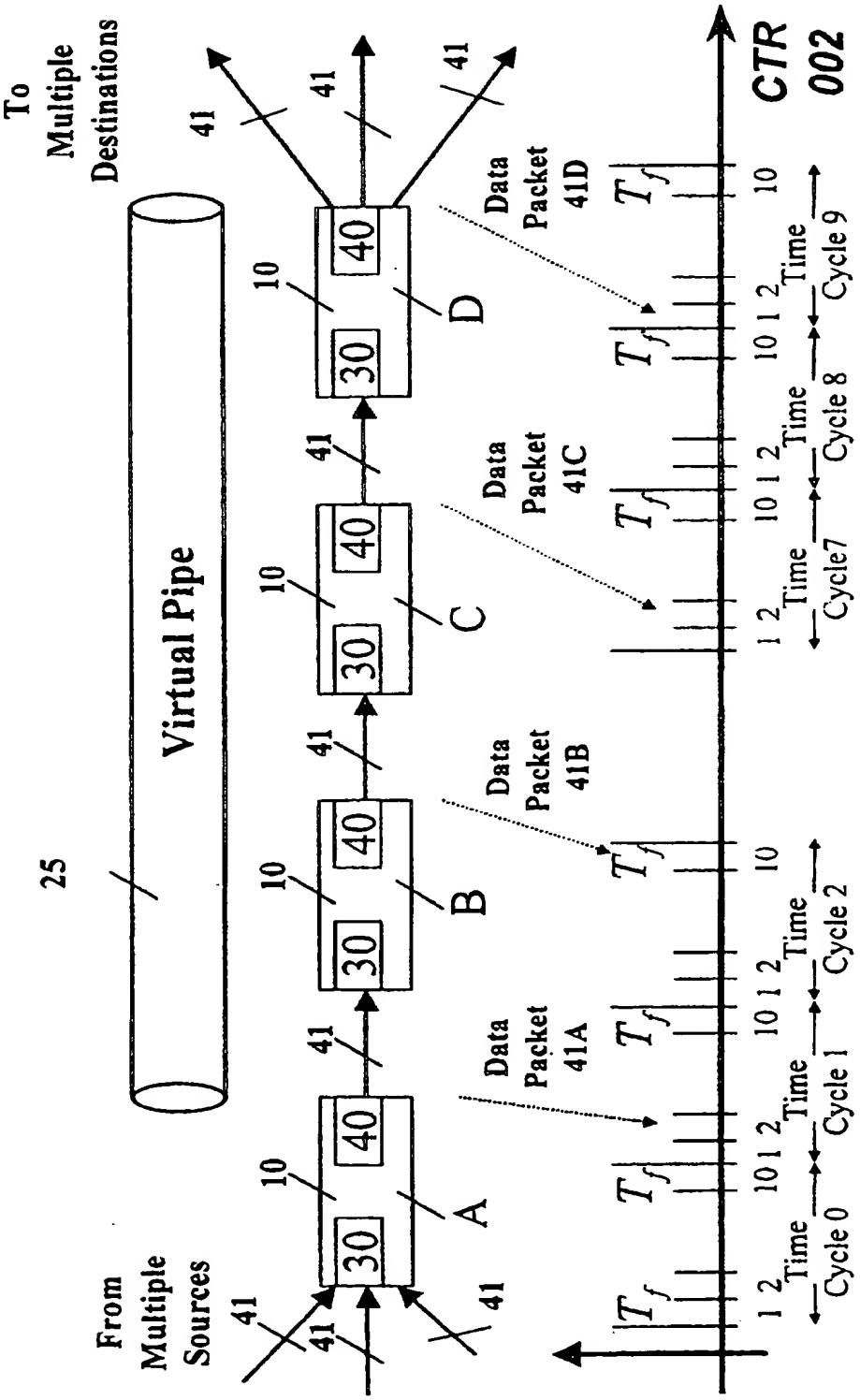
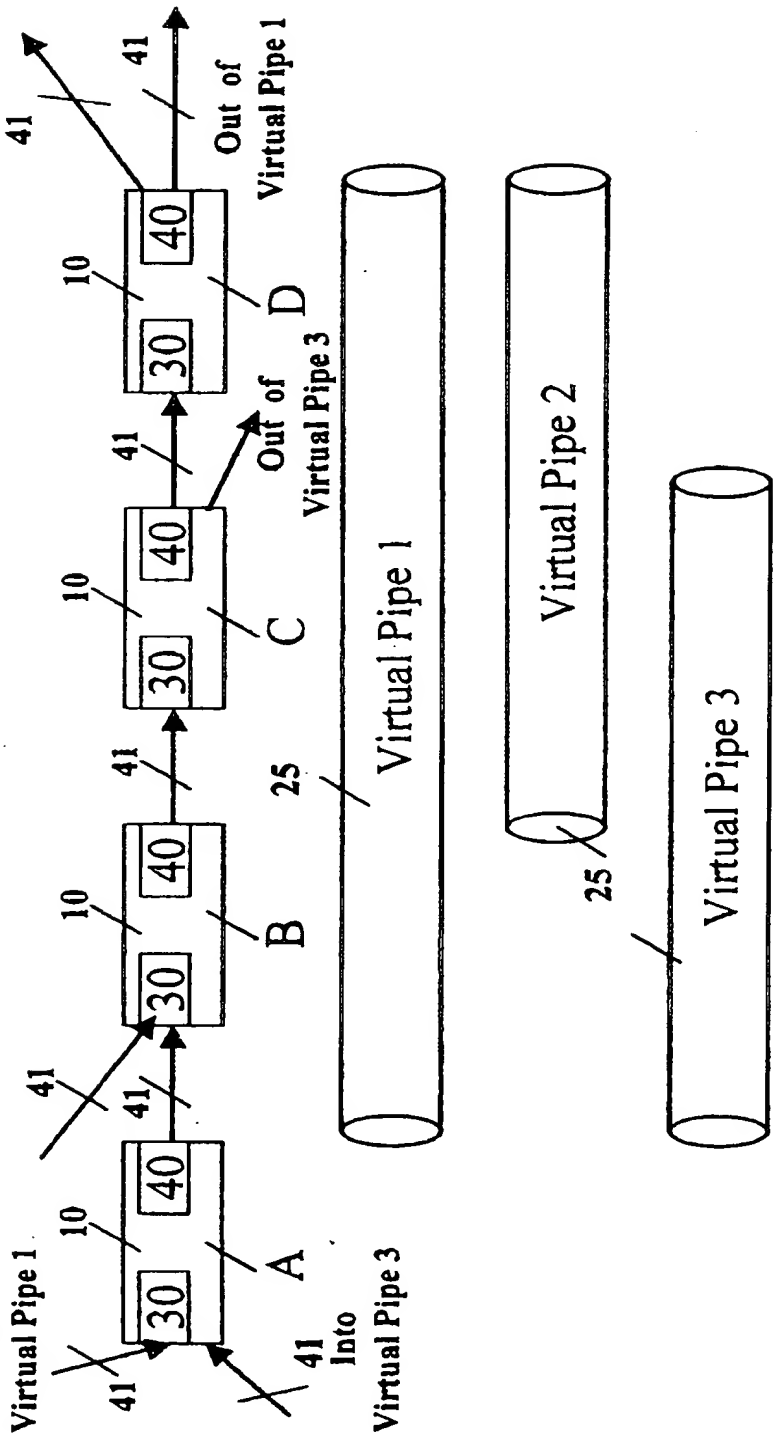

From Multiple Sources
To Multiple Destinations
Virtual Pipe
25
A
B
C
D
10
30
40
41
Data Packet 41A
Data Packet 41B
Data Packet 41C
Data Packet 41D
T_f
1 2 Time Cycle 0
10 1 2 Time Cycle 1
10 1 2 Time Cycle 2
1 2 Time Cycle7
10 1 2 Time Cycle 8
10 1 2 Time Cycle 9
CTR 002

FIG. 2


Into Virtual Pipe 1
Into Virtual Pipe 2
Into Virtual Pipe 3
Out of Virtual Pipe 1
Out of Virtual Pipe 2
Out of Virtual Pipe 3
A
B
C
D
10
25
30
40
41
Virtual Pipe 1
Virtual Pipe 2
Virtual Pipe 3

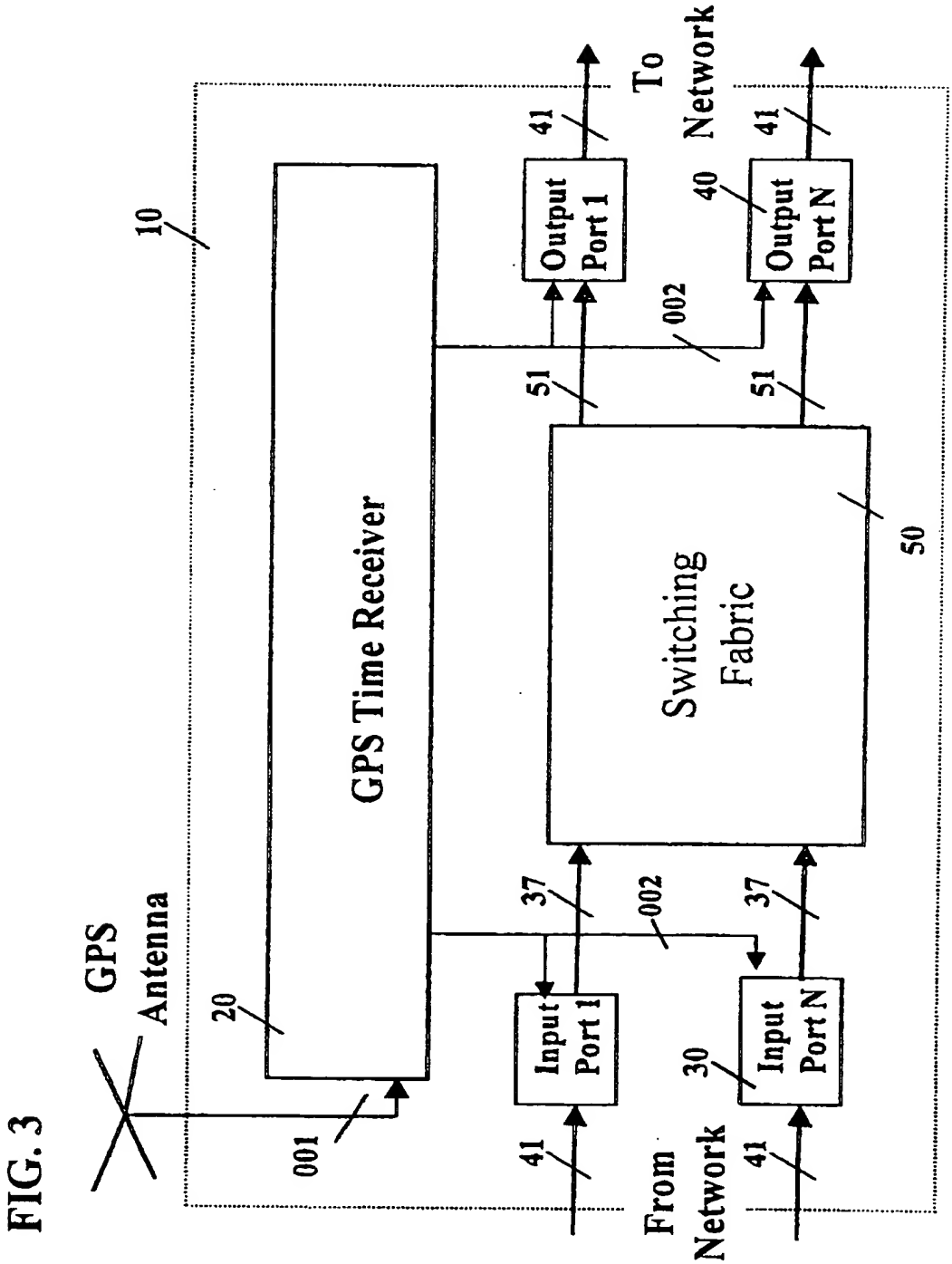

GPS Antenna
10
20
GPS Time Receiver
001
Input Port 1
Input Port N
30
37
002
41
From Network
Switching Fabric
50
51
Output Port 1
Output Port N
40
To Network

FIG. 3

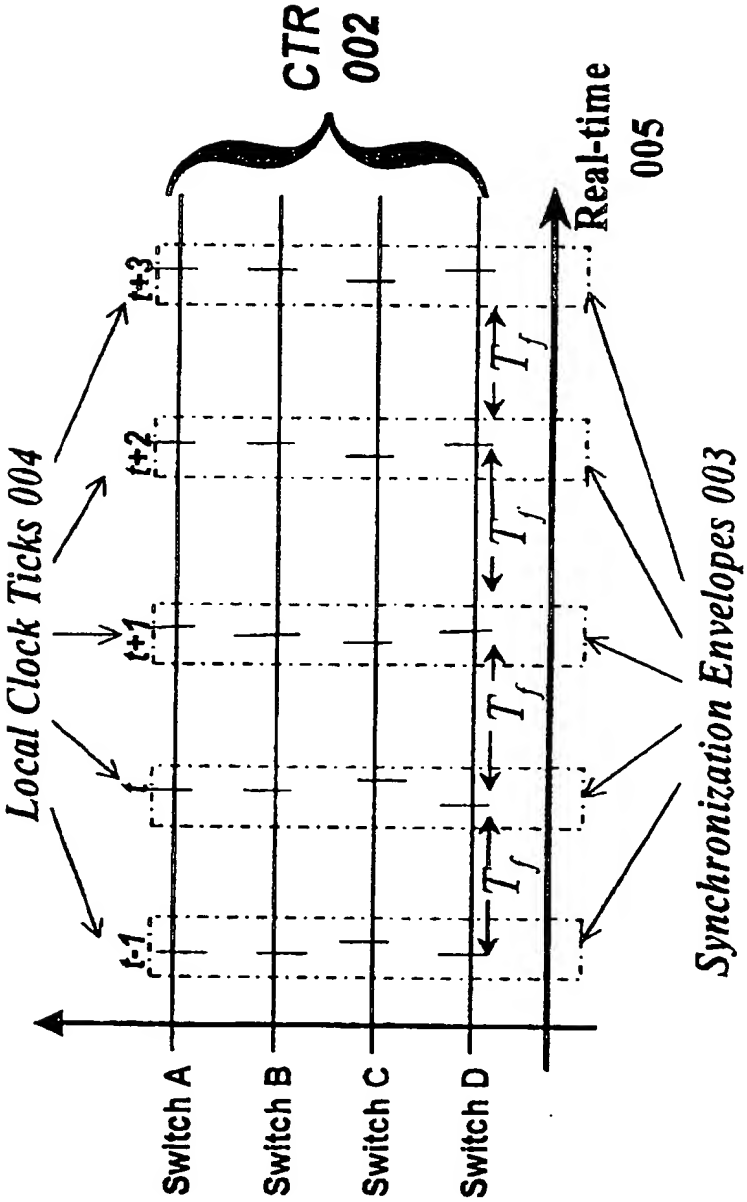

CTR
002
Real-time
005
Local Clock Ticks 004
Synchronization Envelopes 003
t+3
t+2
t+1
t
t-1
T_f
T_f
T_f
T_f
Switch A
Switch B
Switch C
Switch D

FIG. 4

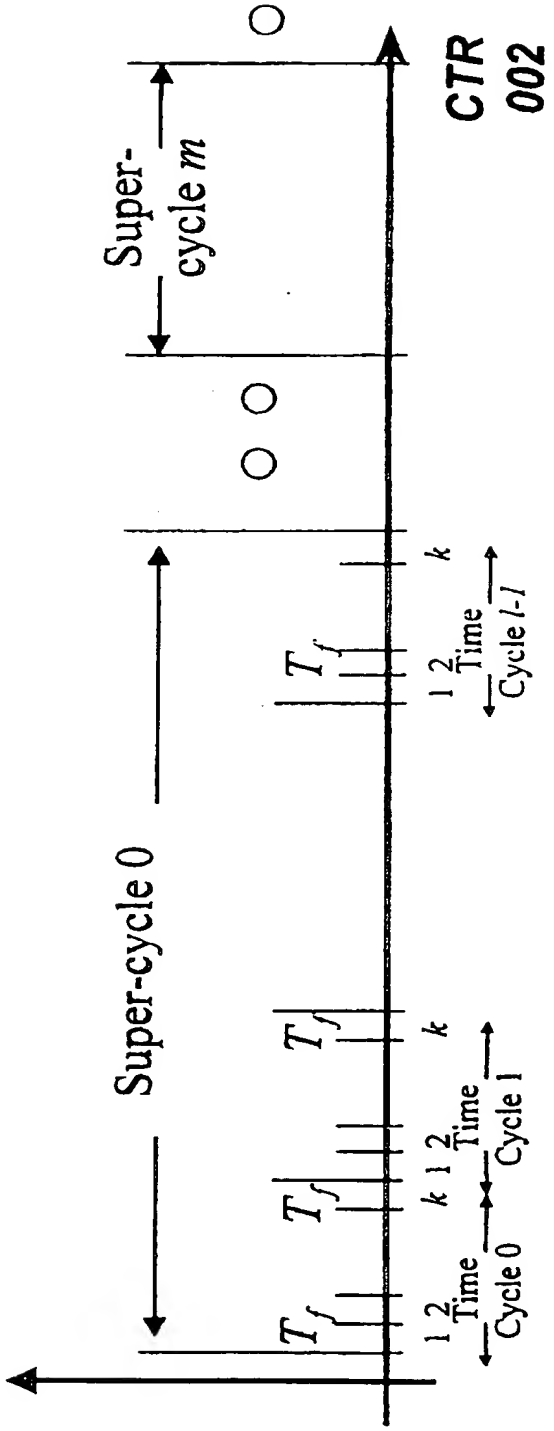
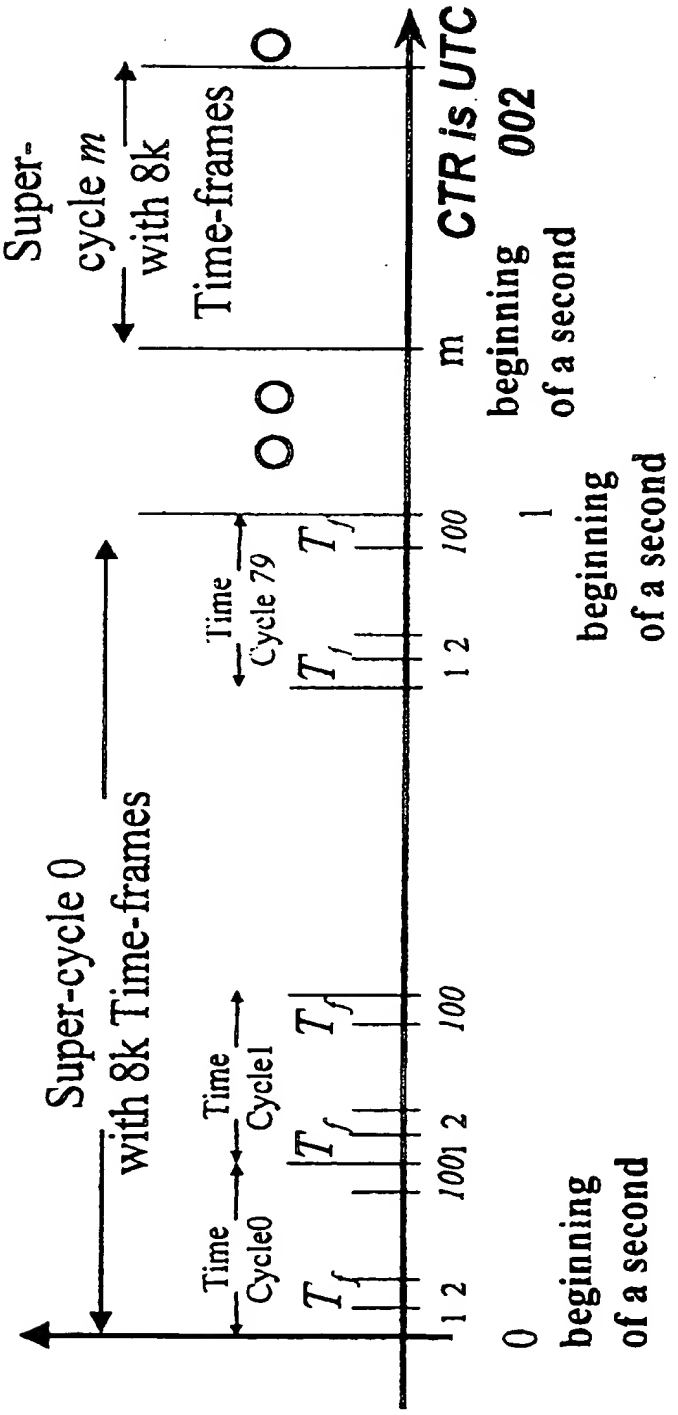

CTR
002
Super-
cycle m
Super-cycle 0
T_f
T_f
T_f
k
k
k
1 2
Time
Cycle 0
1 2
Time
Cycle 1
1 2
Time
Cycle l-1

FIG. 5

FIG. 6


Super-cycle 0
with 8k Time-frames
Super-cycle m with 8k Time-frames
Time Cycle0
Time Cycle1
Time Cycle 79
T_f
1 2 100
0 beginning of a second
1 beginning of a second
m beginning of a second
CTR is UTC 002

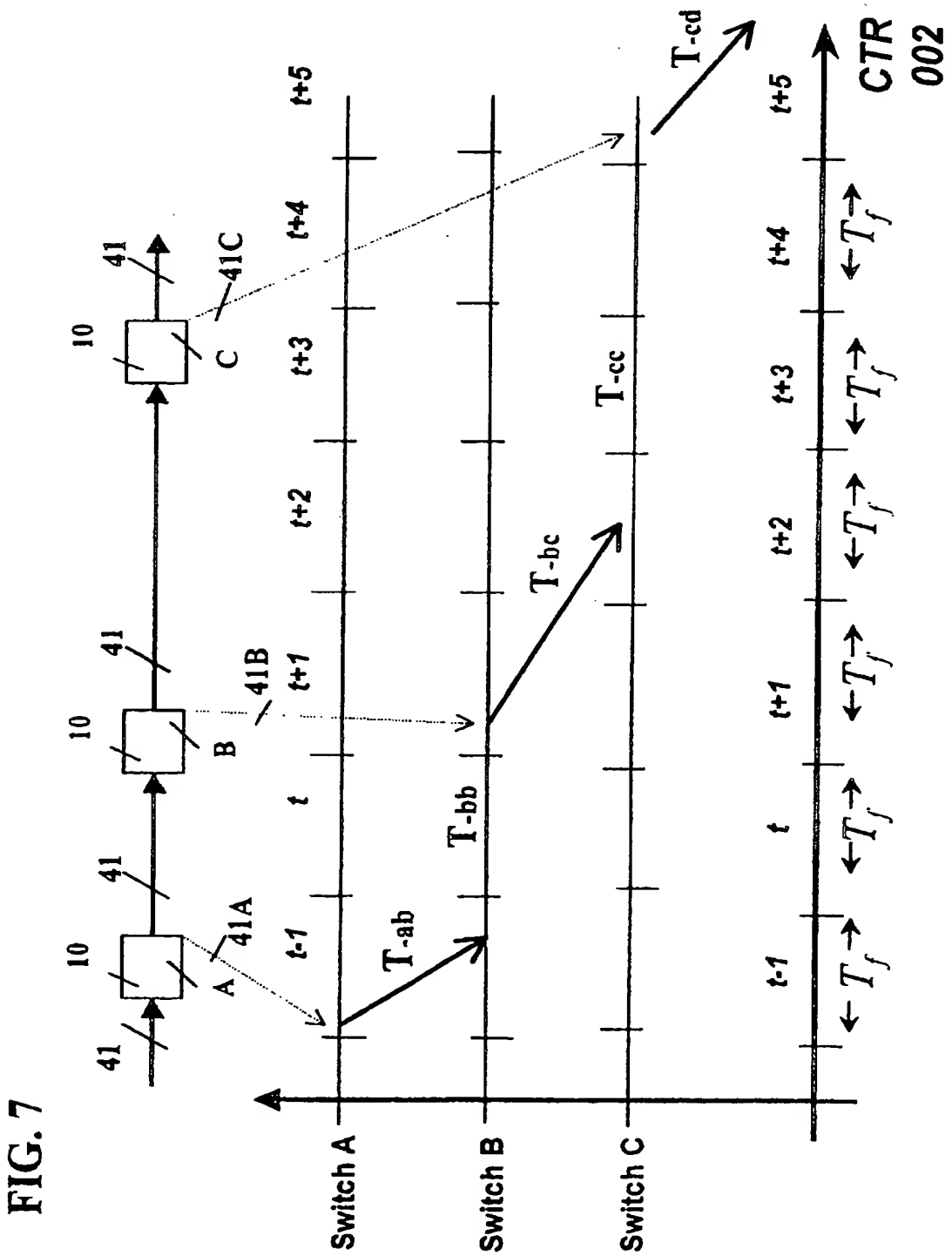
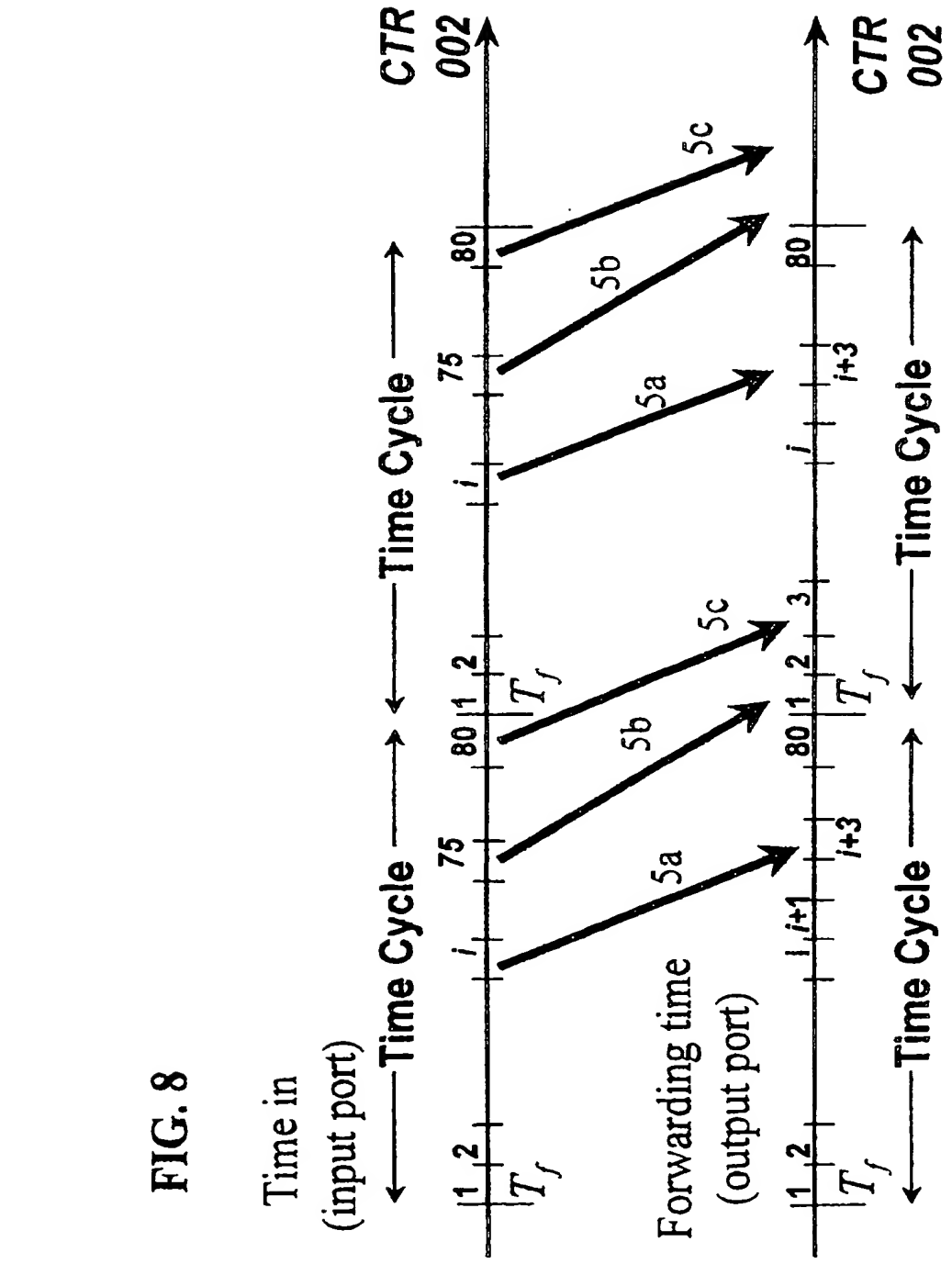

41
10
41
41A
A
10
41
41B
B
10
41
41C
C
t-1
t
t+1
t+2
t+3
t+4
t+5
T-ab
T-bb
T-bc
T-cc
T-cd
Switch A
Switch B
Switch C
CTR
002

FIG. 7

FIG. 8


Time in
(input port)
Time Cycle
CTR
002
1
2
T_f
80
75
i
5a
5b
5c
Forwarding time
(output port)
i+1
i+3
3

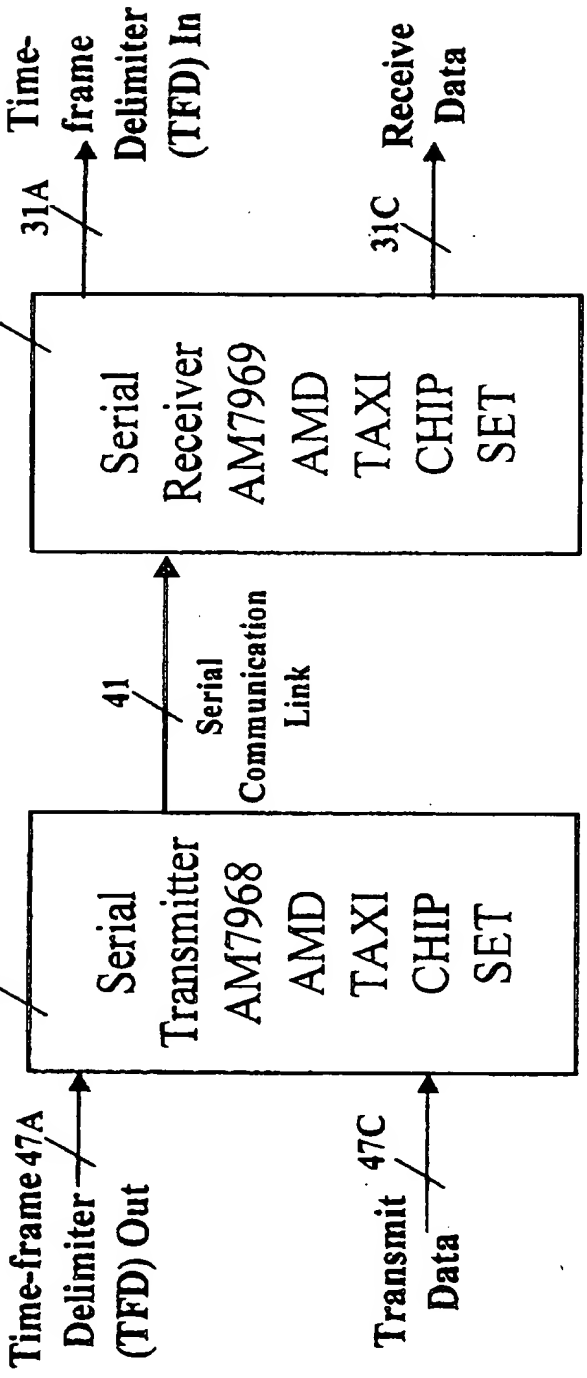

Time-frame Delimiter (TFD) Out
47A
Transmit Data
47C
49
Serial Transmitter AM7968 AMD TAXI CHIP SET
41
Serial Communication Link
31
Serial Receiver AM7969 AMD TAXI CHIP SET
31A
Time-frame Delimiter (TFD) In
31C
Receive Data

FIG. 9

FIG. 10

4B/5B encoding scheme		
HEX DATA	4-bit Binary Data	5-bit Encoded Data Codeword
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
A	1010	10110
B	1011	10111
C	1100	11010
D	1101	11011
E	1110	11100
F	1111	11101

FIG. 11

4B/5B encoding scheme		
Control Input		10-bit Encoded Control Codeword
HEX DATA	Binary Data	
1	0001	11111 11111
2	0010	01101 01101
3	0011	01101 11001
4	0100	11111 00100
5	0101	01101 00111
6	0110	11001 00111
7	0111	11001 11001
8	1000	00100 00100
9	1001	00100 11111
A	1010	00100 00000
B	1011	00111 00111
C	1100	00111 11001
D	1101	00000 00100
E	1110	00000 11111
F	1111	00000 00000

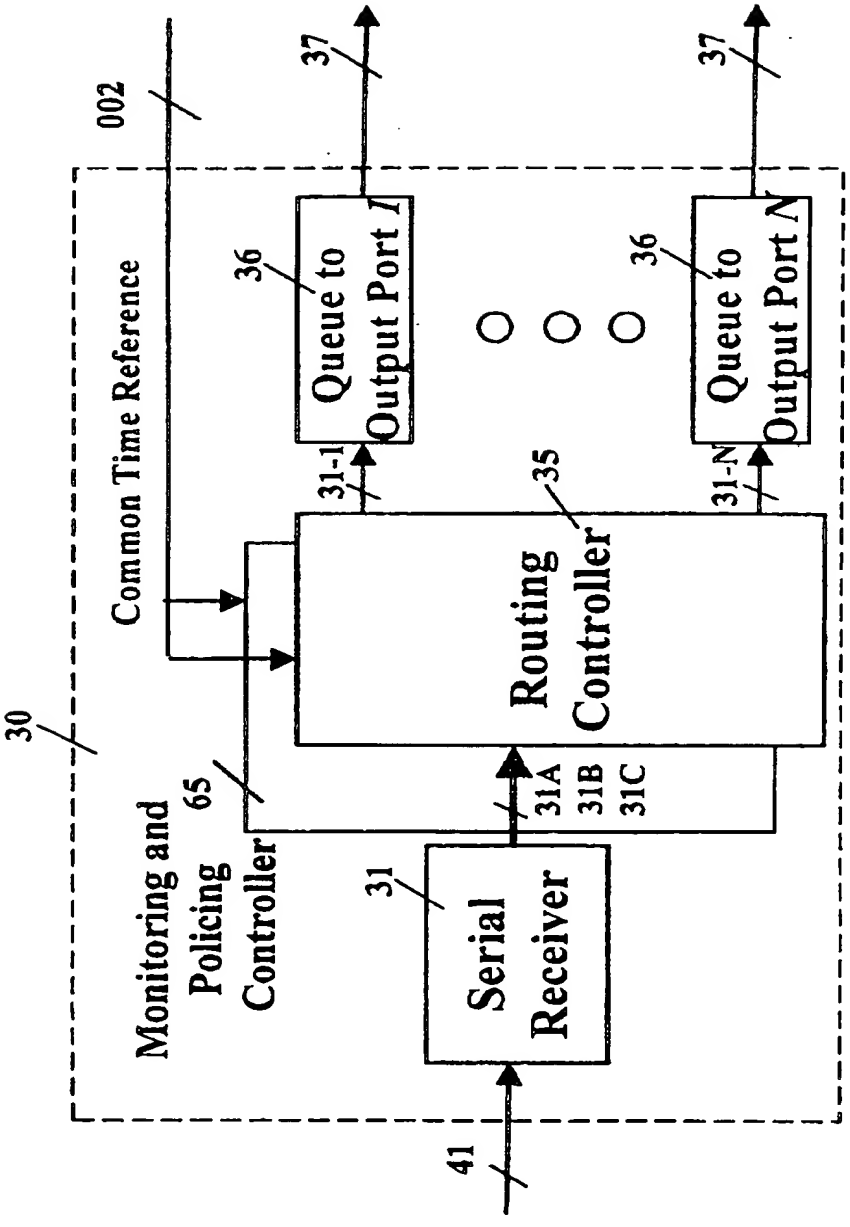

002
Common Time Reference
30
Monitoring and Policing Controller
65
Serial Receiver
31
Routing Controller
35
31A
31B
31C
Queue to Output Port 1
36
Queue to Output Port N
36
31-1
31-N
37
37
41

FIG. 12

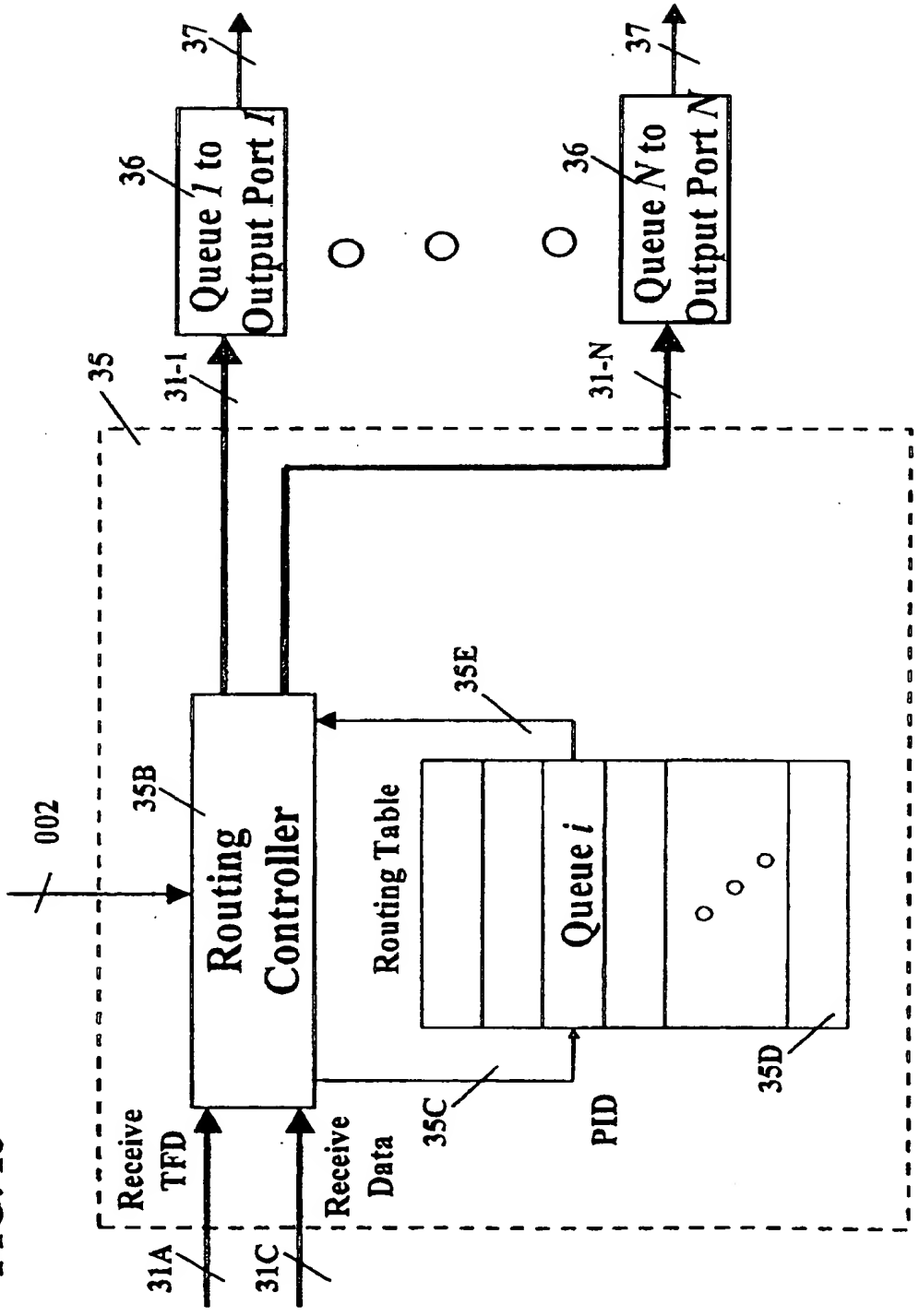

Queue 1 to Output Port 1
Queue N to Output Port N
36
37
35
31-1
31-N
002
35B
Routing Controller
35E
Routing Table
Queue i
35D
35C
PID
Receive TFD
Receive Data
31A
31C

FIG. 13

FIG. 14

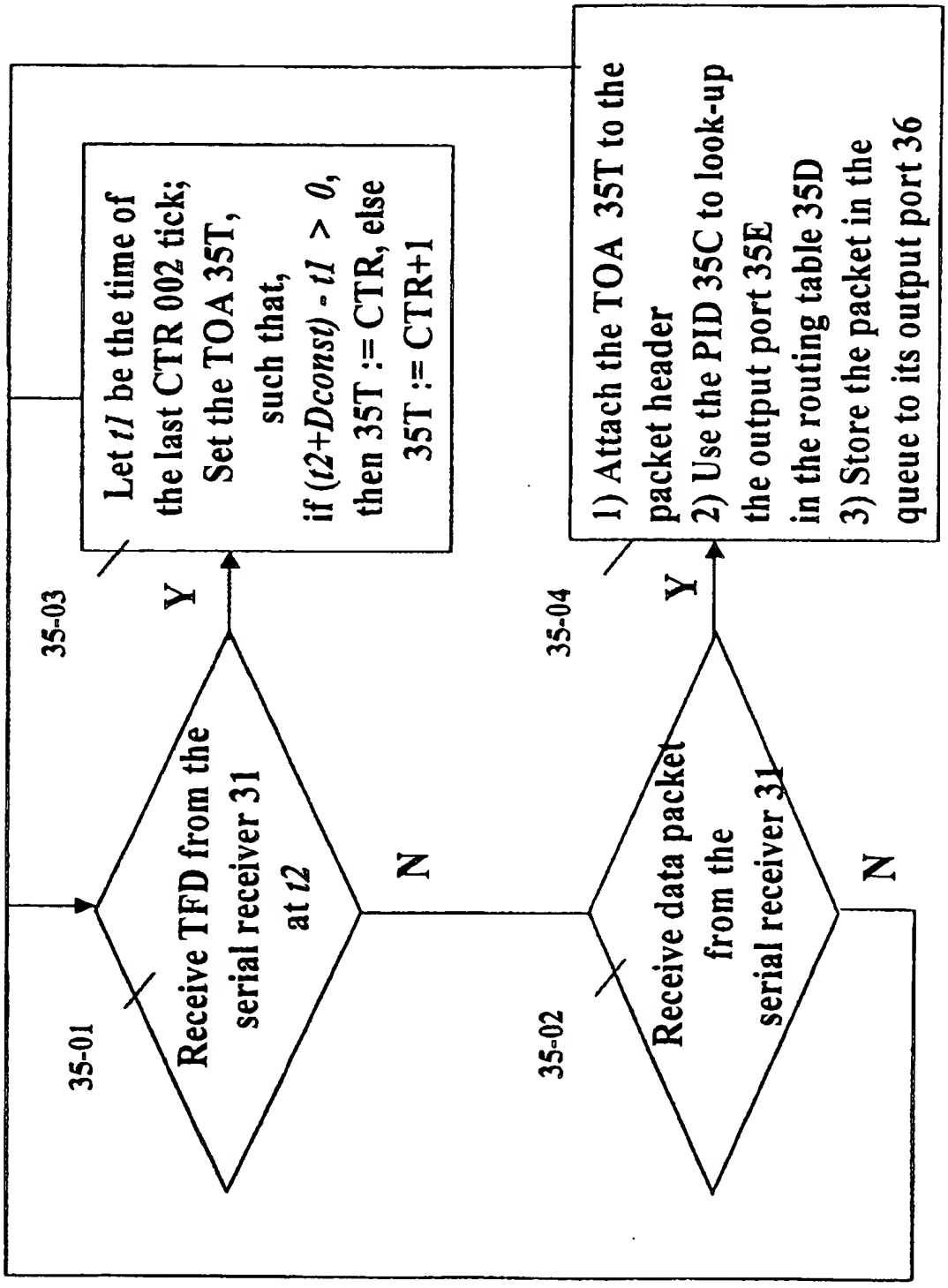

35-01
Receive TFD from the serial receiver 31 at t2
Y
N
35-03
Let t1 be the time of the last CTR 002 tick; Set the TOA 35T, such that, if (t2+Dconst) - t1 > 0, then 35T := CTR, else 35T := CTR+1
35-02
Receive data packet from the serial receiver 31
Y
N
35-04
1) Attach the TOA 35T to the packet header
2) Use the PID 35C to look-up the output port 35E in the routing table 35D
3) Store the packet in the queue to its output port 36

FIG. 15

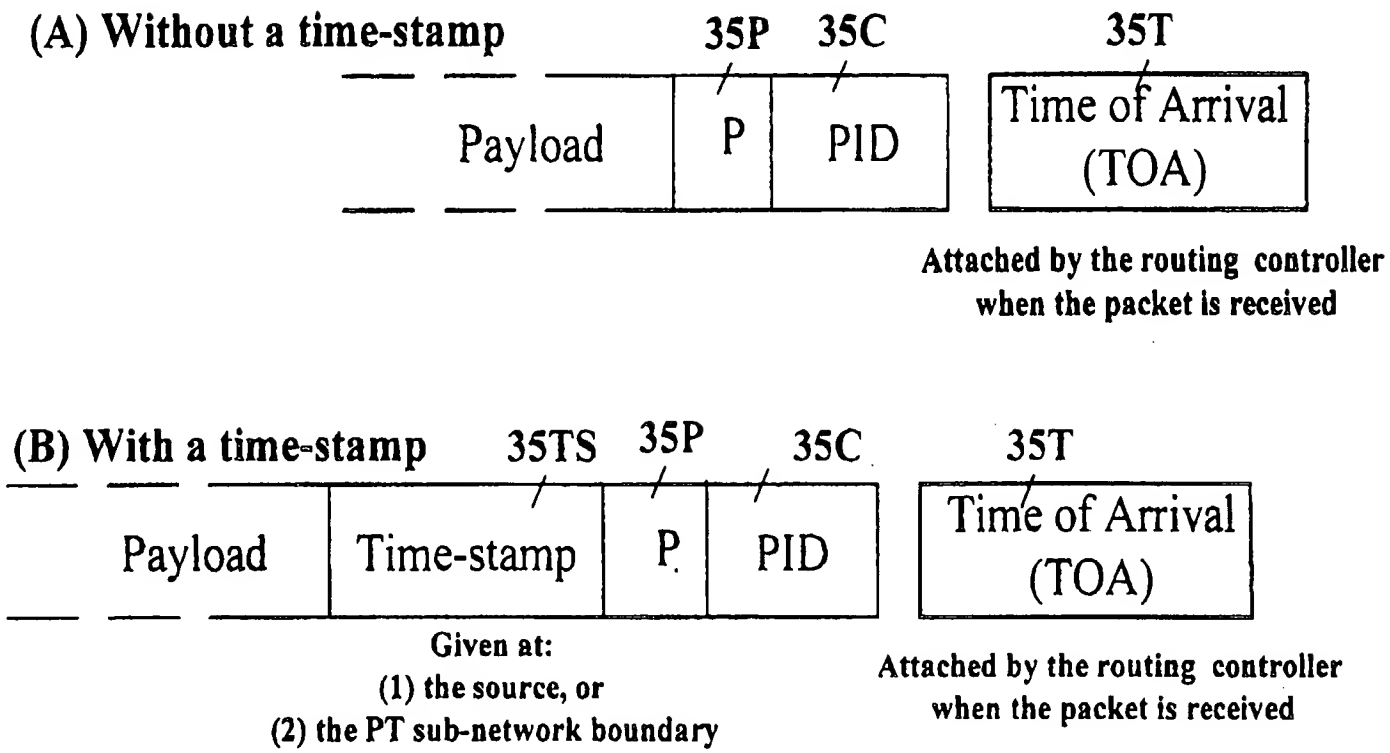
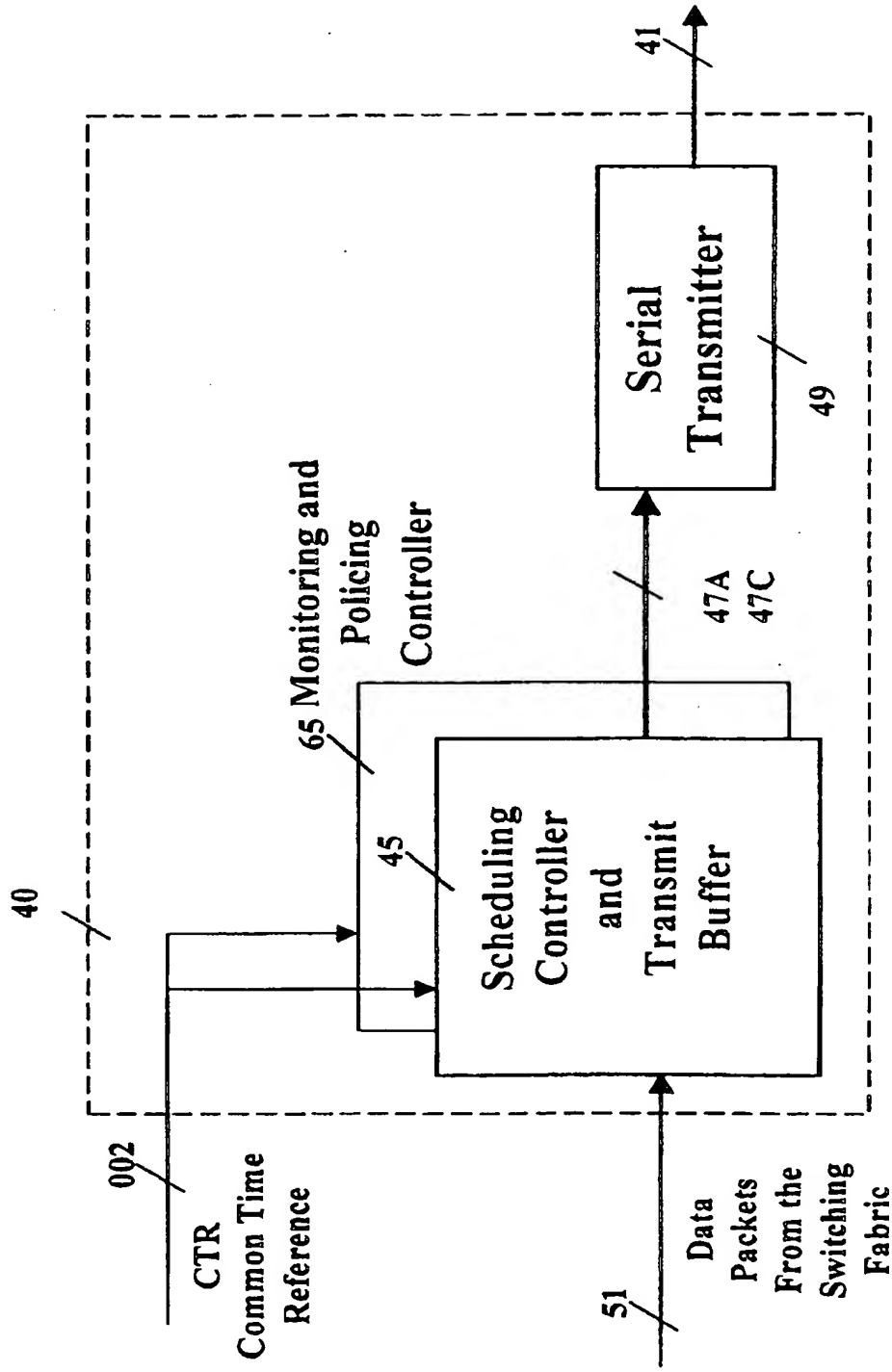

(A) Without a time-stamp
35P
35C
35T
Payload
P
PID
Time of Arrival (TOA)
Attached by the routing controller when the packet is received
(B) With a time-stamp
35TS
35P
35C
35T
Payload
Time-stamp
P
PID
Time of Arrival (TOA)
Given at:
(1) the source, or
(2) the PT sub-network boundary
Attached by the routing controller when the packet is received

FIG. 16


40
002
CTR
Common Time
Reference
51
Data
Packets
From the
Switching
Fabric
65 Monitoring and
Policing
Controller
45
Scheduling
Controller
and
Transmit
Buffer
47A
47C
Serial
Transmitter
49
41

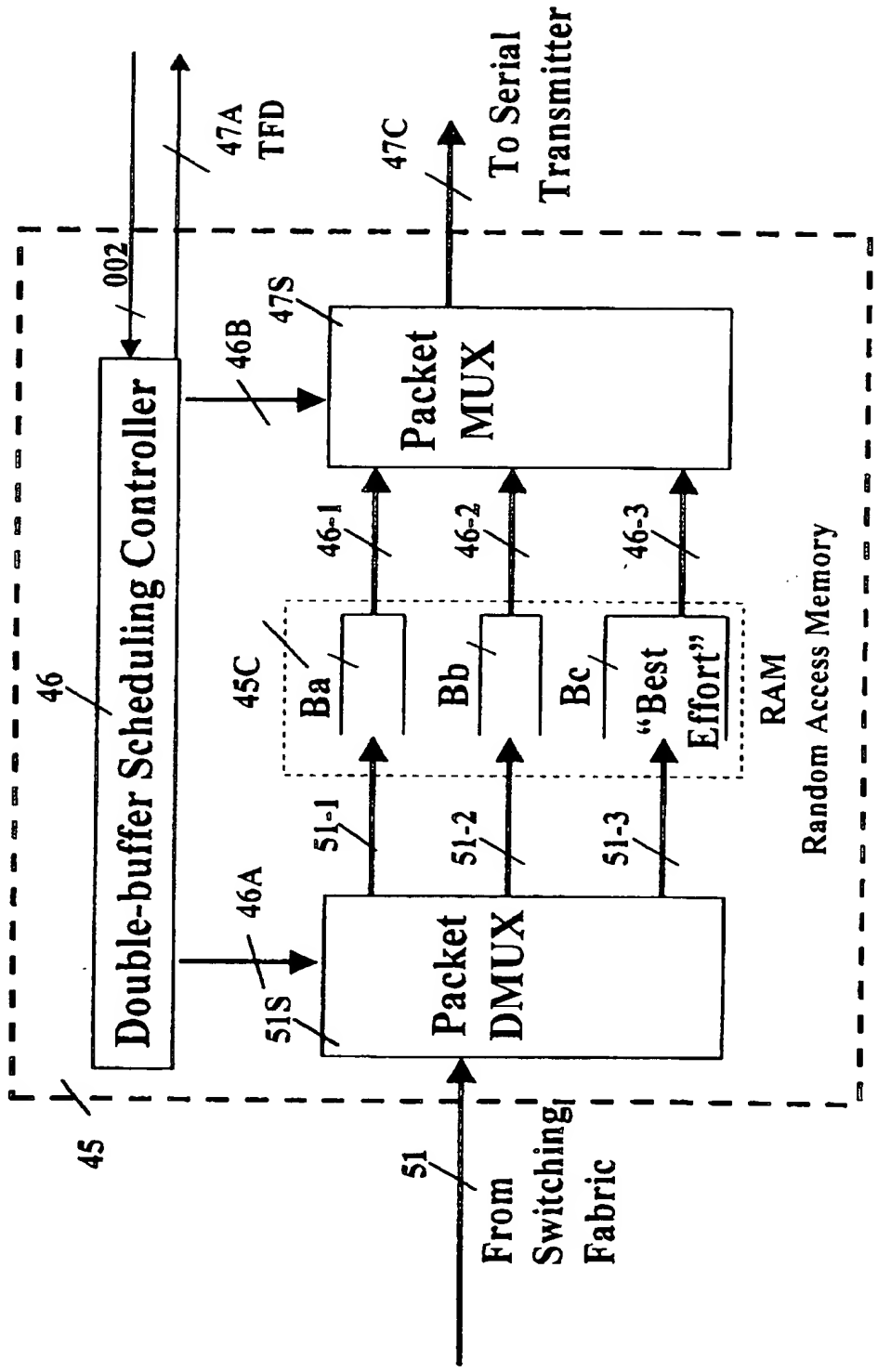

45
51
From Switching Fabric
51S
Packet DMUX
46A
51-1
51-2
51-3
45C
Ba
Bb
Bc
"Best Effort"
RAM
Random Access Memory
46
Double-buffer Scheduling Controller
002
46-1
46-2
46-3
46B
47S
Packet MUX
47A TFD
47C
To Serial Transmitter

FIG. 17

FIG. 18

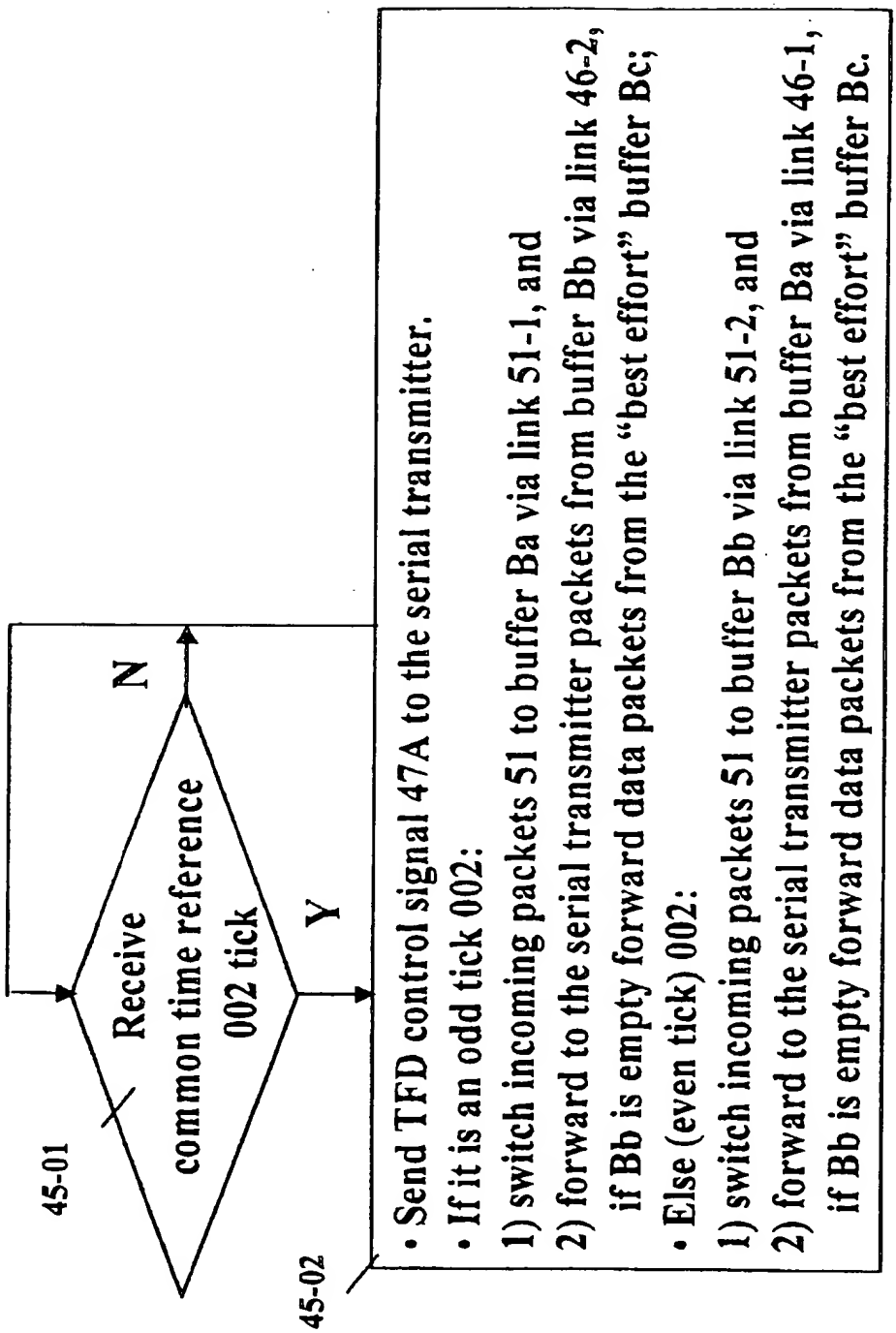

45-01
Receive common time reference 002 tick
N
Y
45-02
• Send TFD control signal 47A to the serial transmitter.
• If it is an odd tick 002:
1) switch incoming packets 51 to buffer Ba via link 51-1, and
2) forward to the serial transmitter packets from buffer Bb via link 46-2, if Bb is empty forward data packets from the "best effort" buffer Bc;
• Else (even tick) 002:
1) switch incoming packets 51 to buffer Bb via link 51-2, and
2) forward to the serial transmitter packets from buffer Ba via link 46-1, if Bb is empty forward data packets from the "best effort" buffer Bc.

FIG. 19

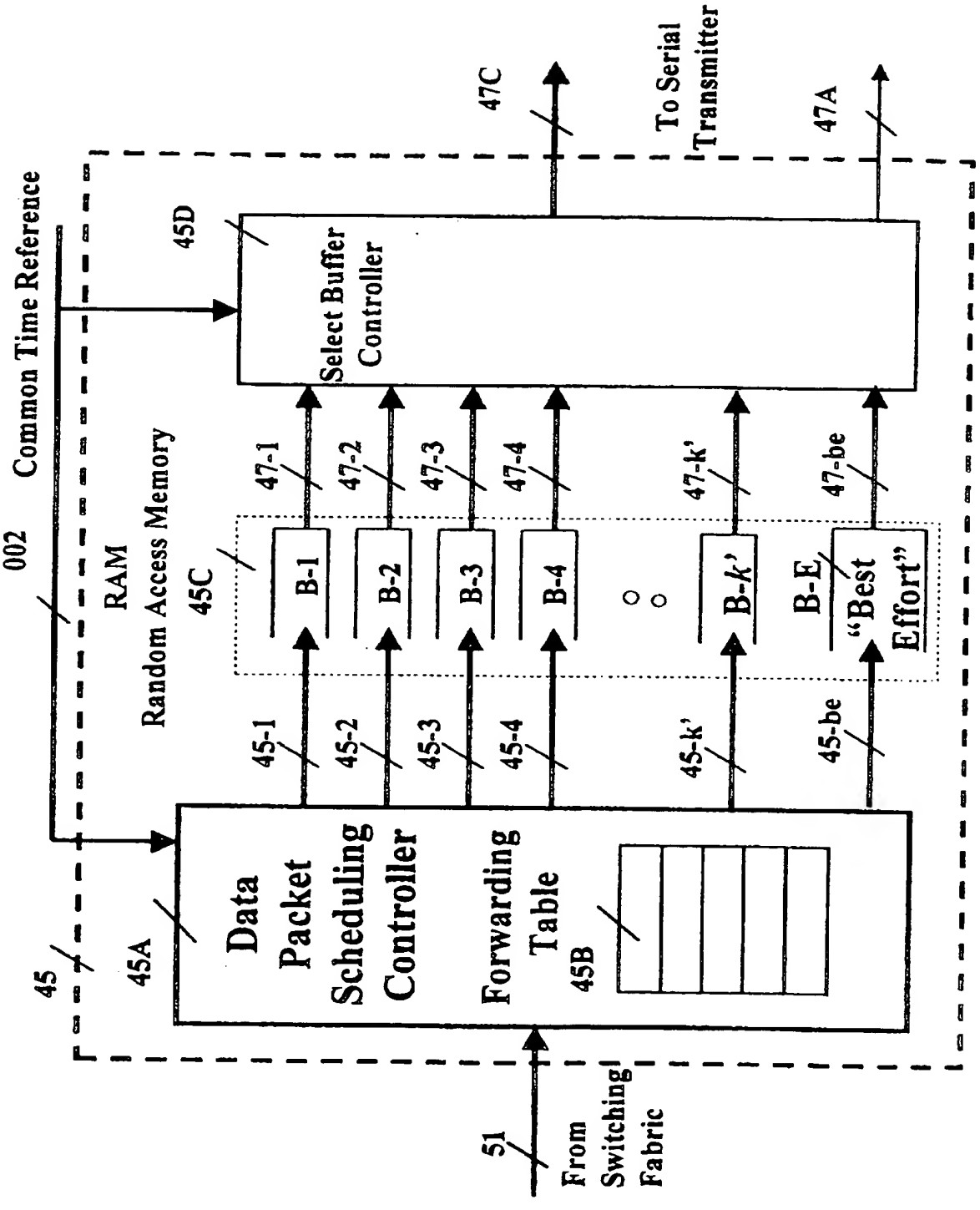

45
45A
Data Packet Scheduling Controller
Forwarding Table
45B
002
Common Time Reference
RAM
Random Access Memory
45C
45D
Select Buffer Controller
51
From Switching Fabric
45-1
45-2
45-3
45-4
45-k'
45-be
B-1
B-2
B-3
B-4
B-k'
B-E
"Best Effort"
47-1
47-2
47-3
47-4
47-k'
47-be
47C
47A
To Serial Transmitter

FIG. 20

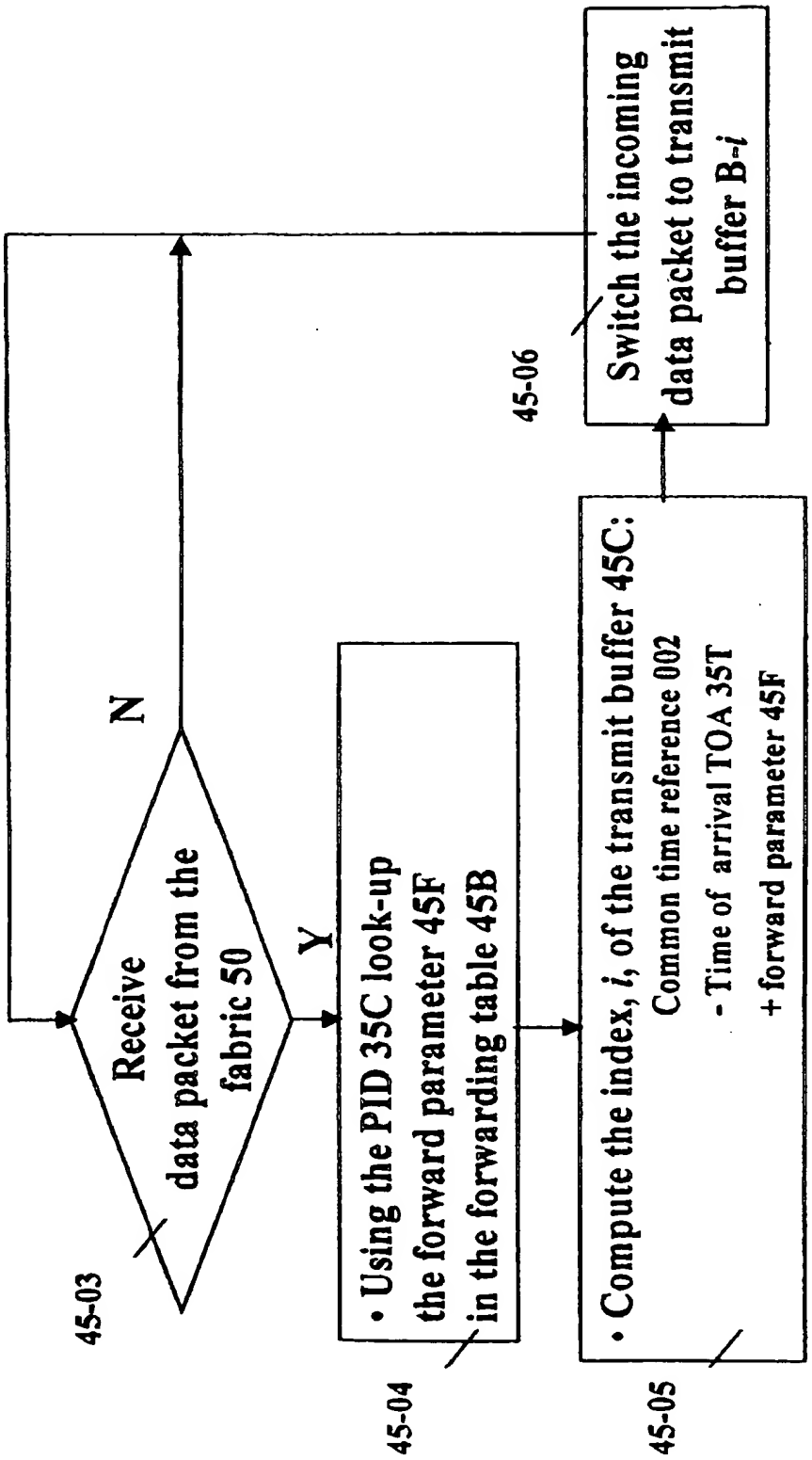

Receive data packet from the fabric 50
45-03
N
Y
• Using the PID 35C look-up the forward parameter 45F in the forwarding table 45B
45-04
• Compute the index, i, of the transmit buffer 45C:
Common time reference 002
- Time of arrival TOA 35T
+ forward parameter 45F
45-05
Switch the incoming data packet to transmit buffer B-i
45-06

FIG. 21

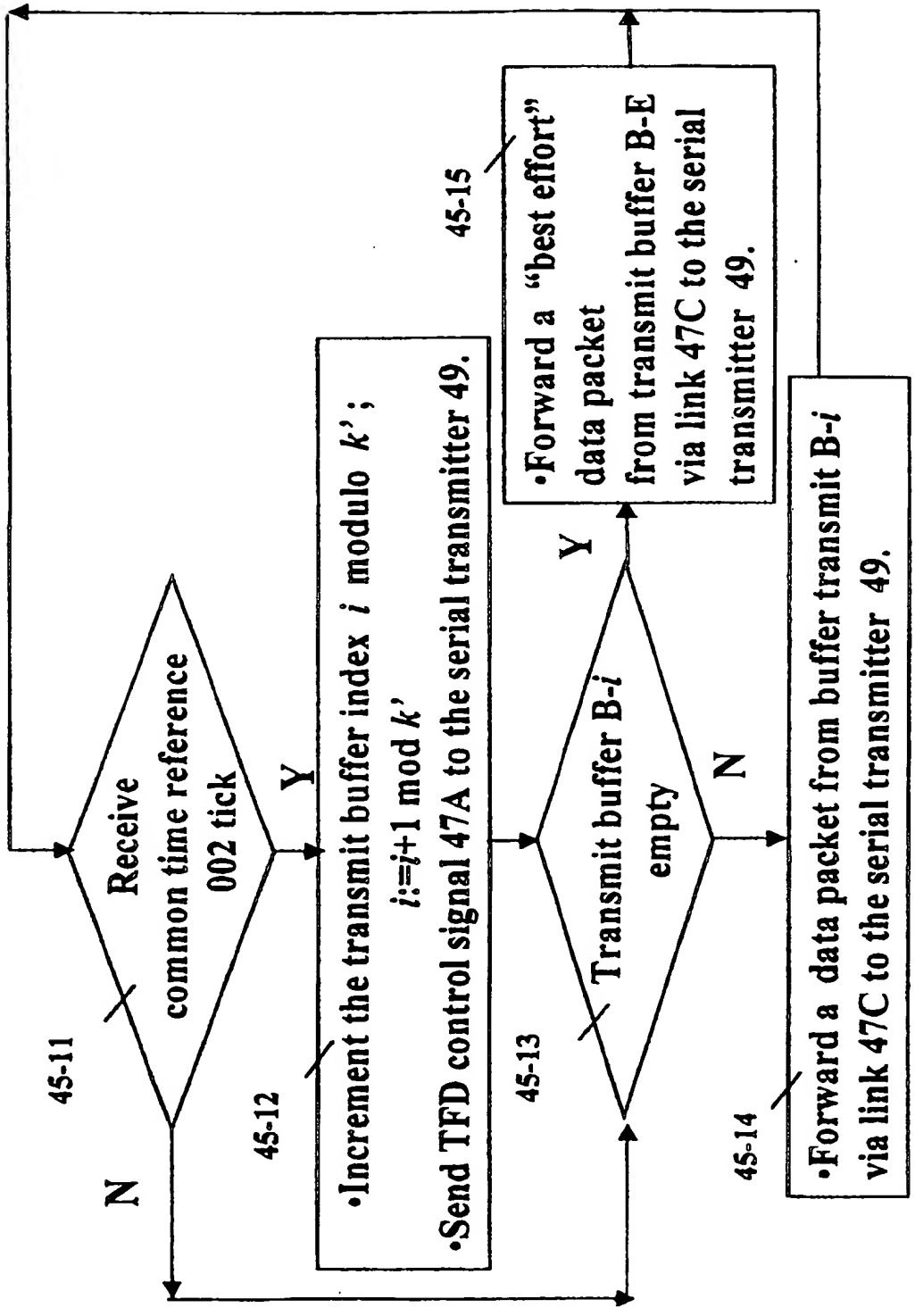
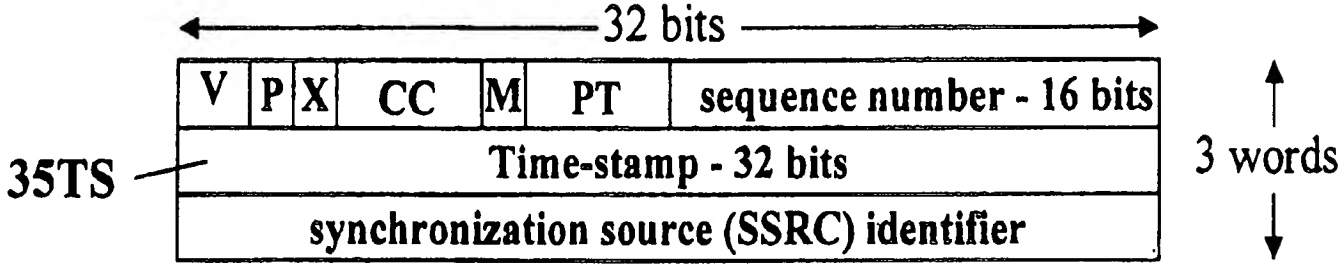

45-11
Receive common time reference 002 tick
N
Y
45-12
•Increment the transmit buffer index i modulo k' ; i:=i+1 mod k'
•Send TFD control signal 47A to the serial transmitter 49.
45-13
Transmit buffer B-i empty
Y
45-15
•Forward a "best effort" data packet from transmit buffer B-E via link 47C to the serial transmitter 49.
N
45-14
•Forward a data packet from buffer transmit B-i via link 47C to the serial transmitter 49.

FIG. 22

- Real-time protocol (RTP) with the following fields in the header:
  - version (V) - 2 bits
  - padding (P) - 1 bit
  - extension (X) - 1 bit
  - CSRC count (CC) - 4 bits
  - marker (M) - 1 bit
  - payload type - 7 bits
  - sequence number - 16 bits
  - times-tamp - 32 bits



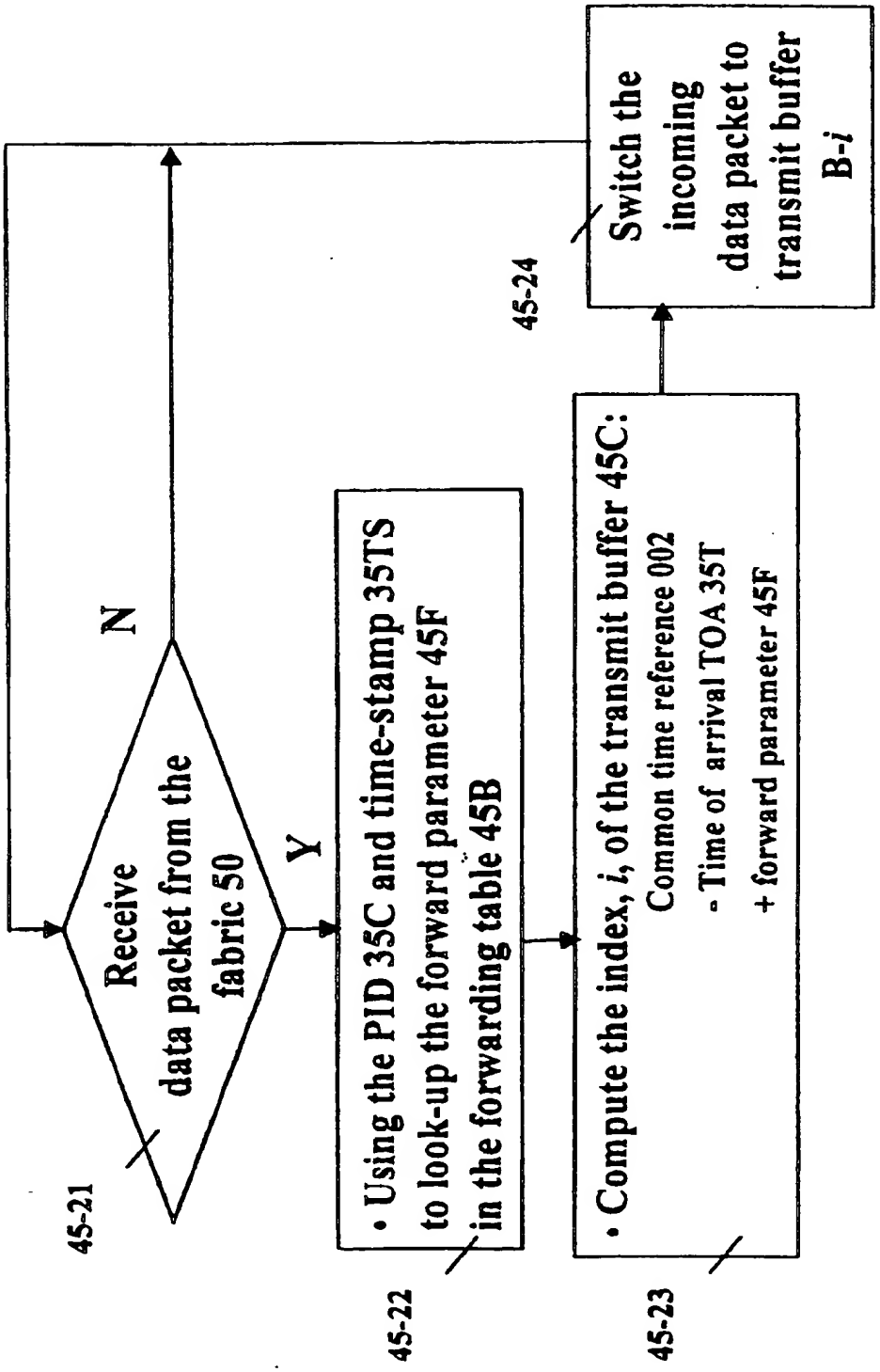

Receive data packet from the fabric 50
45-21
N
Y
• Using the PID 35C and time-stamp 35TS to look-up the forward parameter 45F in the forwarding table 45B
45-22
• Compute the index, i, of the transmit buffer 45C:
Common time reference 002
- Time of arrival TOA 35T
+ forward parameter 45F
45-23
Switch the incoming data packet to transmit buffer B-i
45-24

FIG. 23

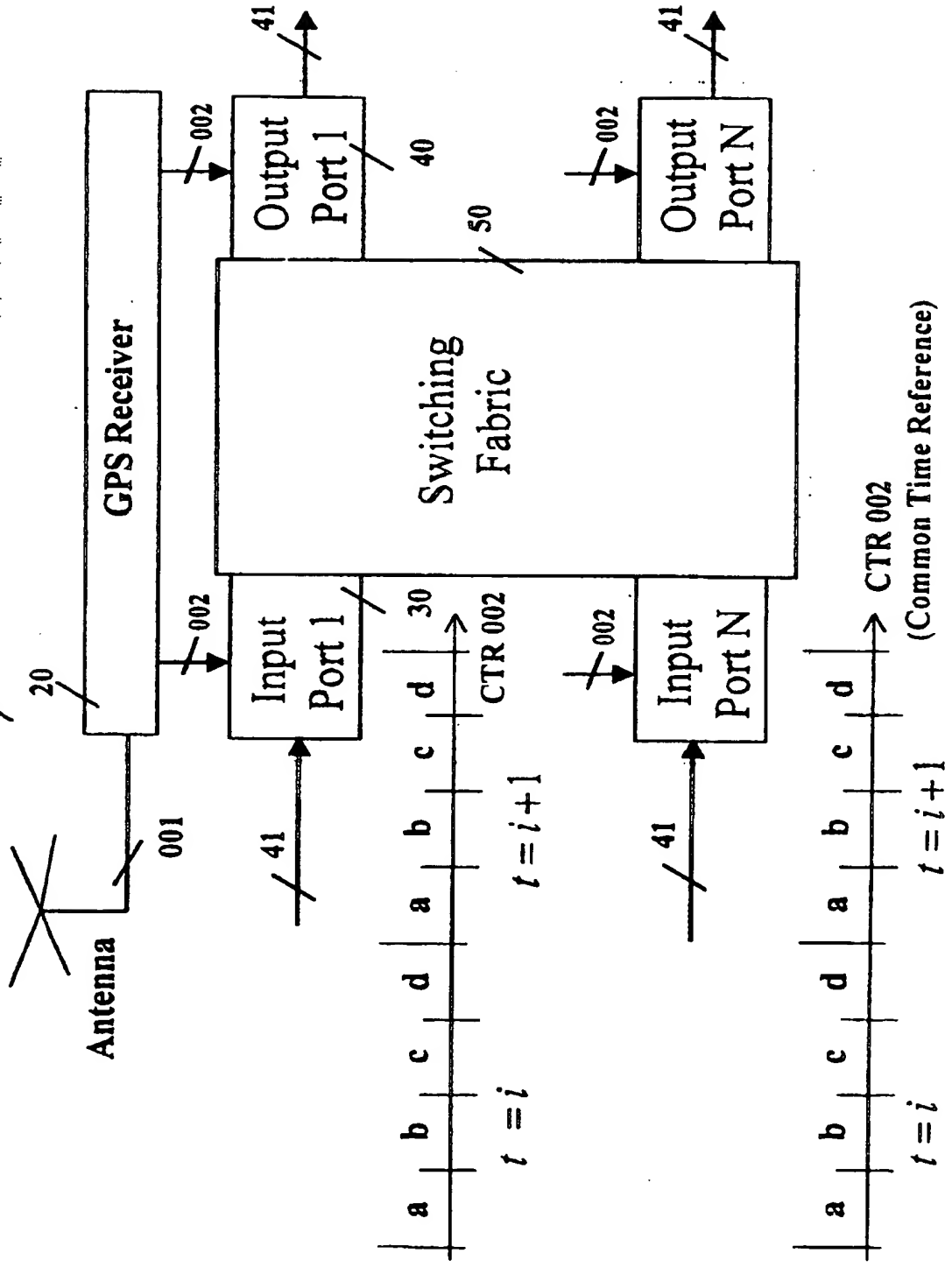

10
20
GPS Receiver
Antenna
001
002
Input Port 1
Output Port 1
Switching Fabric
Input Port N
Output Port N
40
41
50
30
CTR 002
(Common Time Reference)
a
b
c
d
t = i
t = i+1

FIG. 24

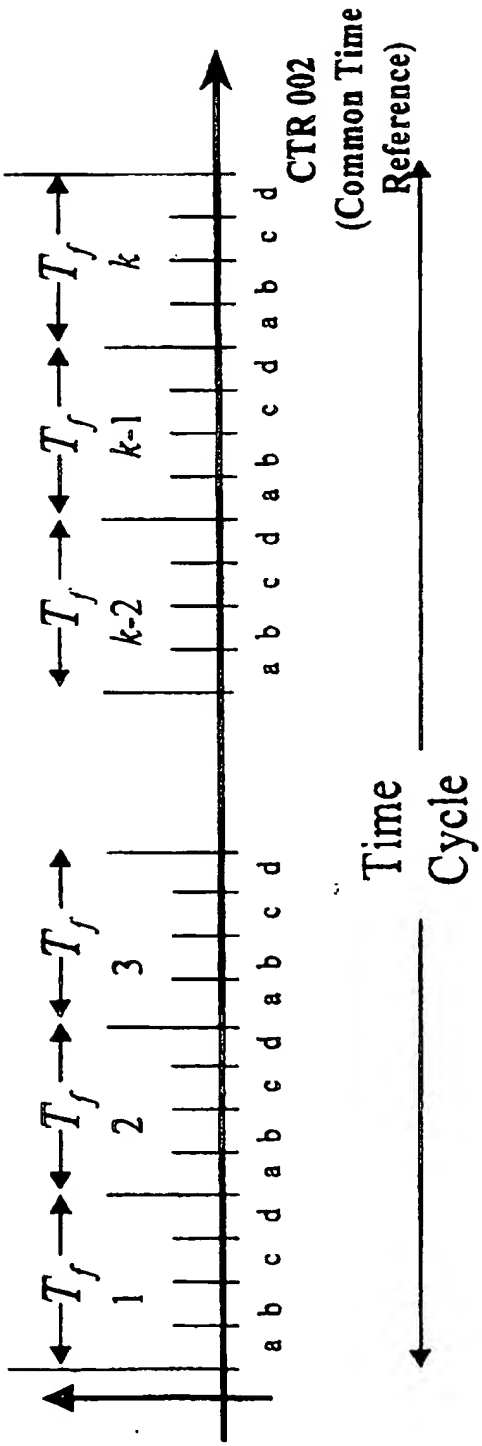

CTR 002
(Common Time Reference)
Time Cycle
T_f
1
2
3
k-2
k-1
k
a b c d a b c d a b c d a b c d a b c d

FIG. 25

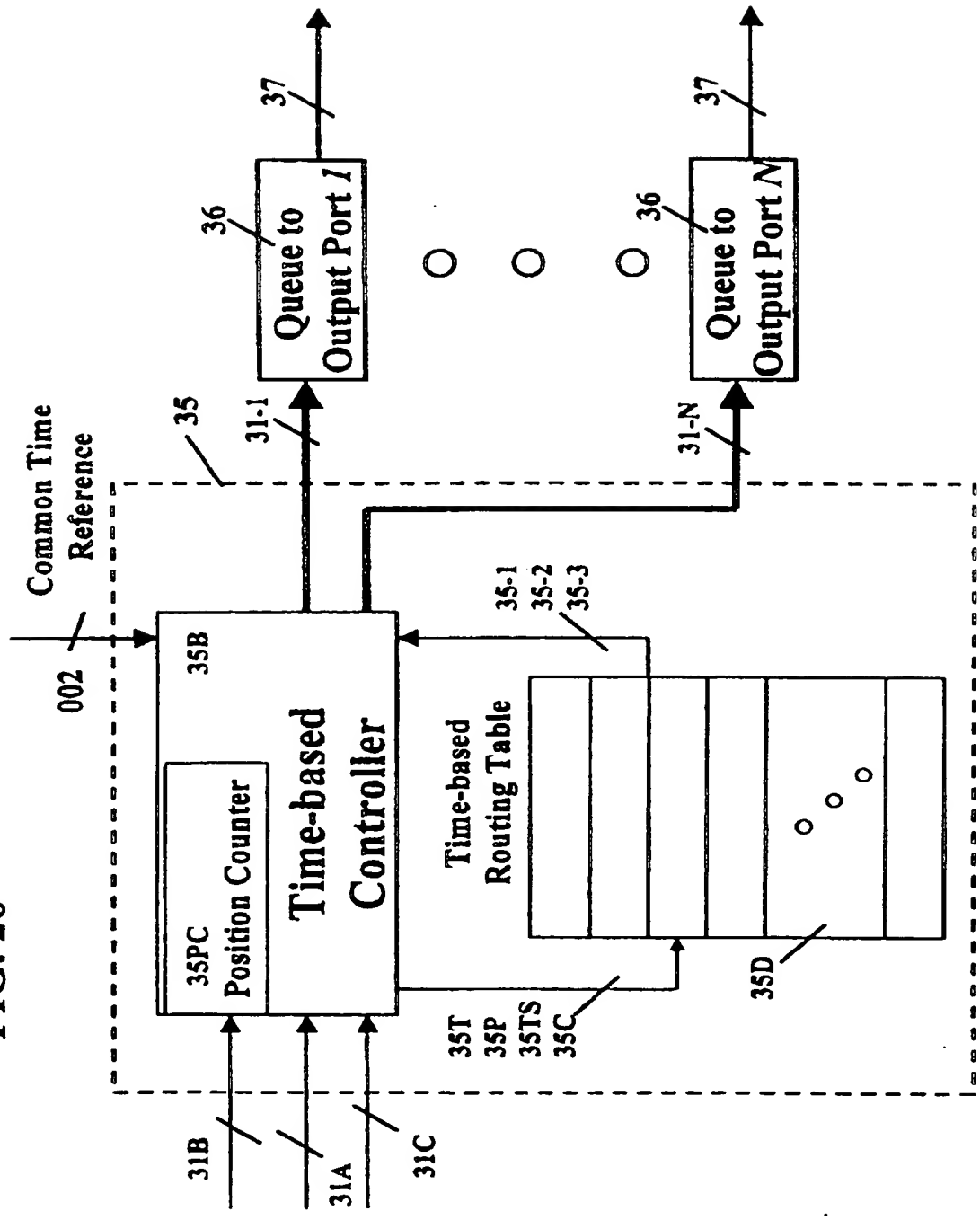

Common Time Reference
002
35
35B
Time-based Controller
35PC Position Counter
Time-based Routing Table
35-1
35-2
35-3
35D
35T
35P
35TS
35C
31B
31A
31C
31-1
31-N
36
Queue to Output Port 1
Queue to Output Port N
37

FIG. 26

FIG. 27

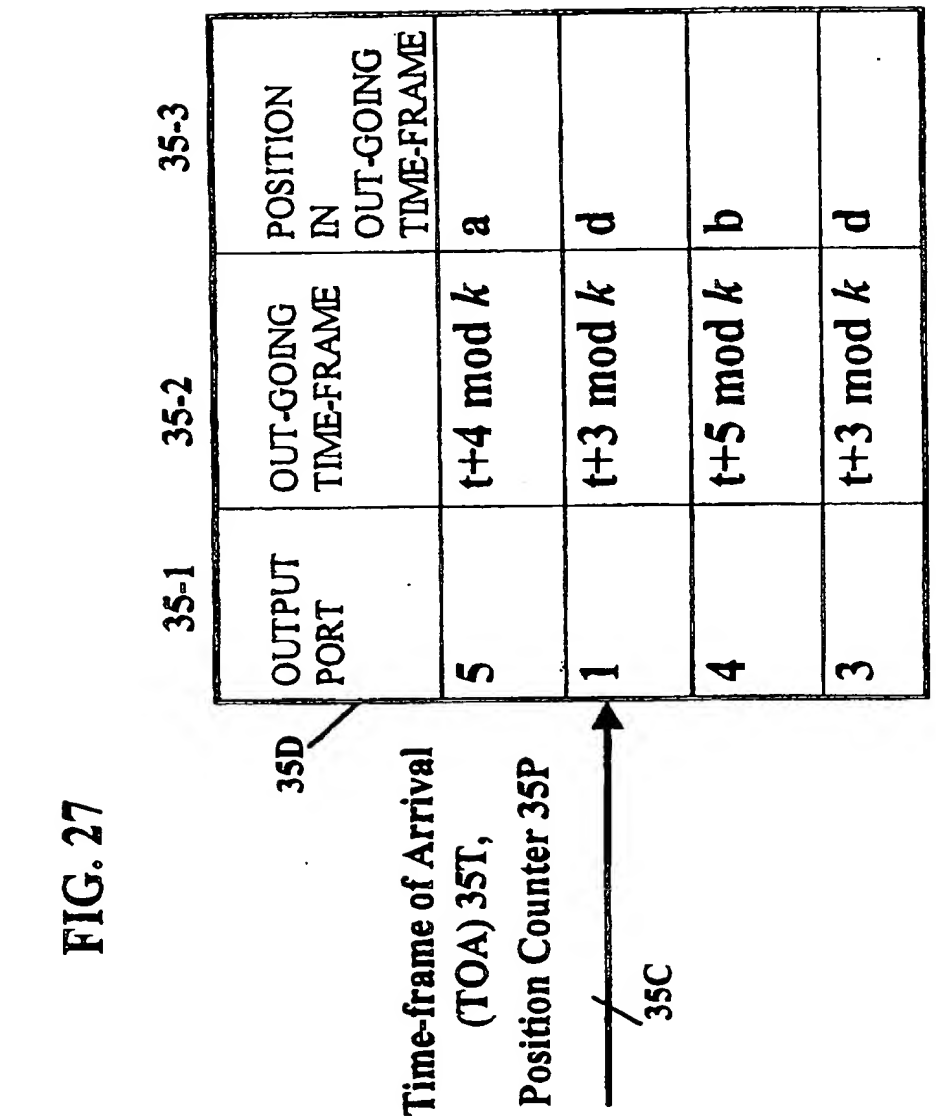

35-1
35-2
35-3
OUTPUT PORT
OUT-GOING TIME-FRAME
POSITION IN OUT-GOING TIME-FRAME
5
t+4 mod k
a
1
t+3 mod k
d
4
t+5 mod k
b
3
t+3 mod k
d
35D
Time-frame of Arrival (TOA) 35T, Position Counter 35P
35C

FIG. 28

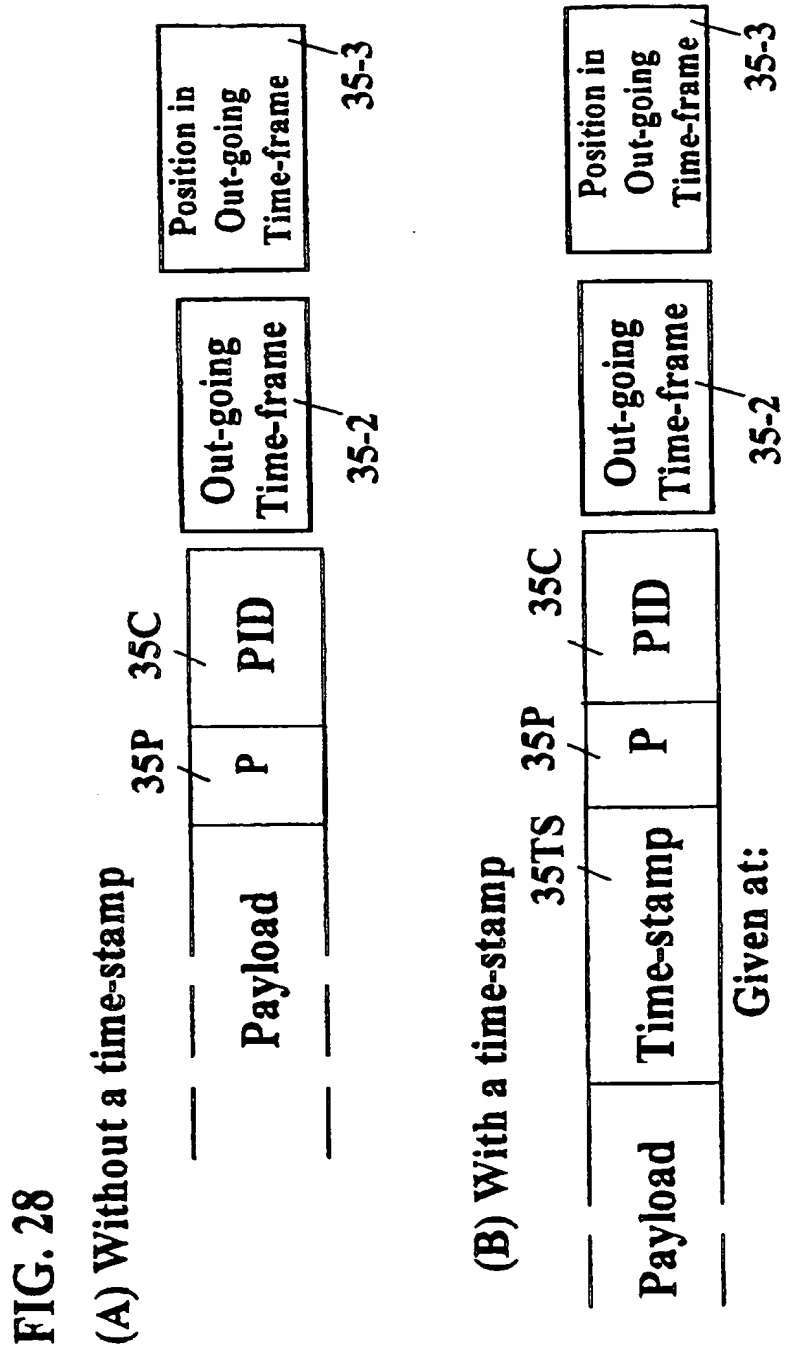

(A) Without a time-stamp
Payload
P
PID
35P
35C
Out-going Time-frame
35-2
Position in Out-going Time-frame
35-3
(B) With a time-stamp
Payload
Time-stamp
P
PID
35TS
35P
35C
Out-going Time-frame
35-2
Position in Out-going Time-frame
35-3
Given at:
(1) the source, or
(2) the PT sub-network boundary

FIG. 29

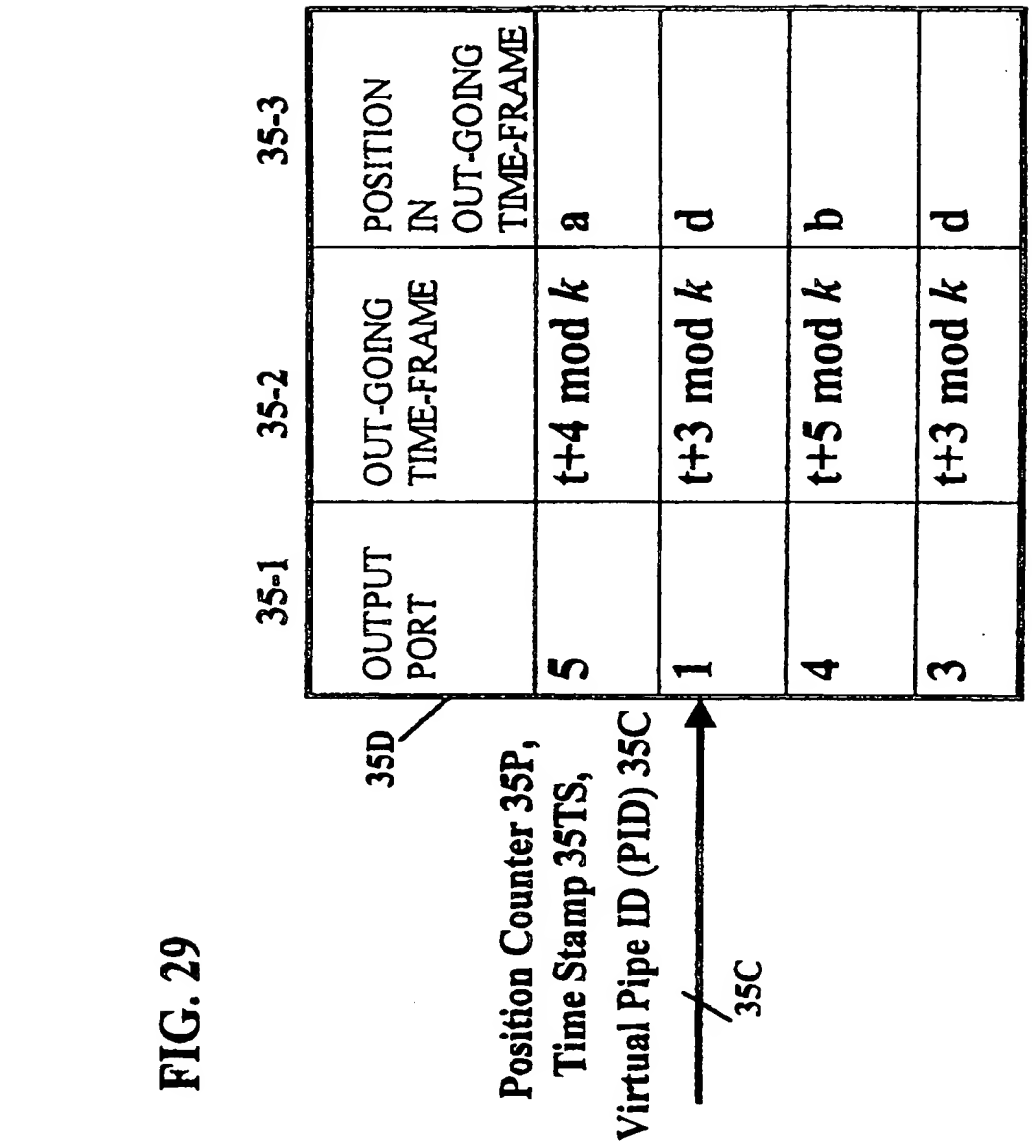
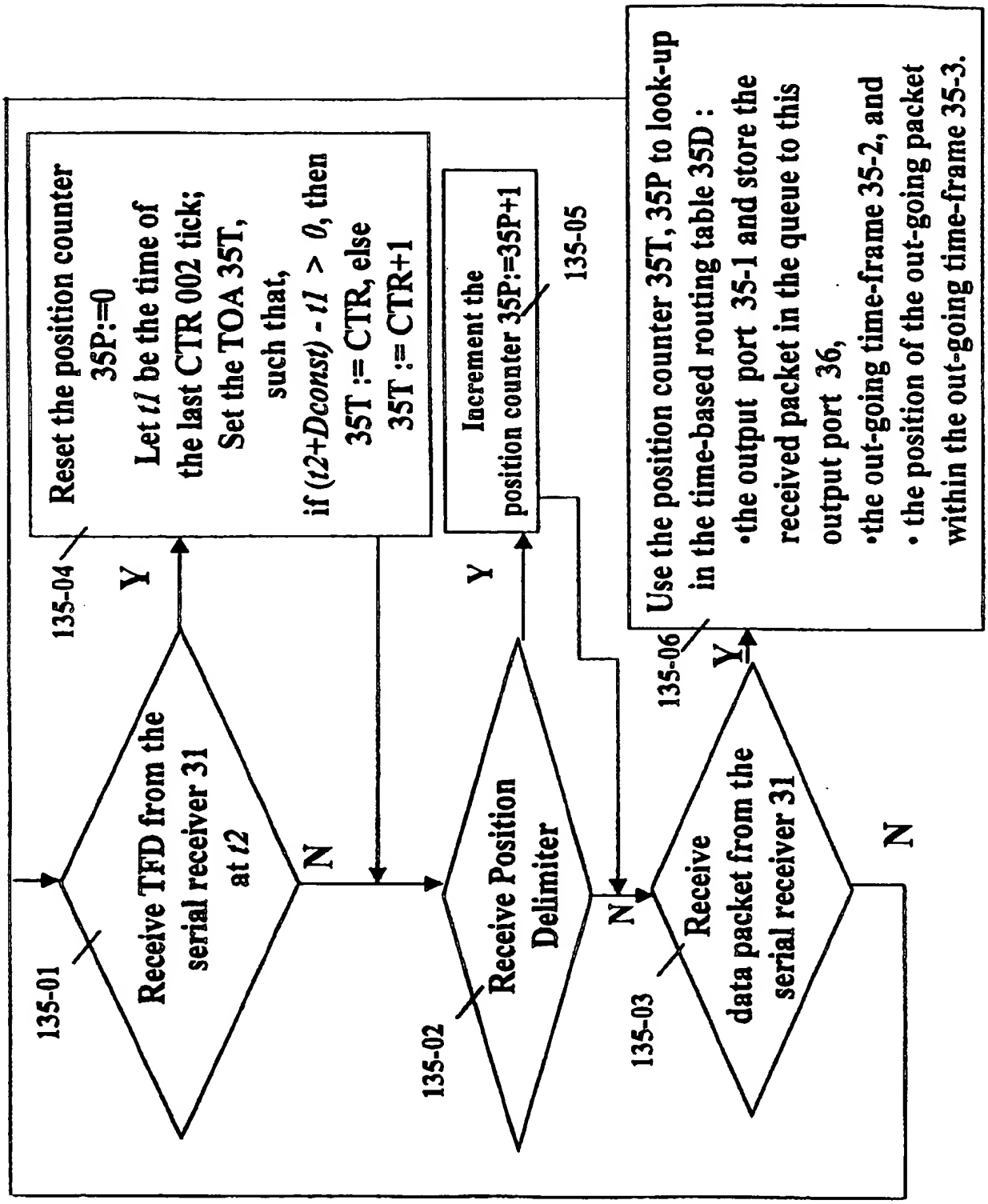

35-1
35-2
35-3
OUTPUT PORT
OUT-GOING TIME-FRAME
POSITION IN OUT-GOING TIME-FRAME
5
t+4 mod k
a
1
t+3 mod k
d
4
t+5 mod k
b
3
t+3 mod k
d
35D
Position Counter 35P, Time Stamp 35TS, Virtual Pipe ID (PID) 35C
35C

FIG. 30


135-01
Receive TFD from the serial receiver 31 at t2
Y
135-04
Reset the position counter 35P:=0
Let t1 be the time of the last CTR 002 tick;
Set the TOA 35T, such that,
if (t2+Dconst) - t1 > 0, then 35T := CTR, else 35T := CTR+1
N
135-02
Receive Position Delimiter
Y
Increment the position counter 35P:=35P+1
135-05
N
135-03
Receive data packet from the serial receiver 31
Y
135-06
Use the position counter 35T, 35P to look-up in the time-based routing table 35D :
•the output port 35-1 and store the received packet in the queue to this output port 36,
•the out-going time-frame 35-2, and
• the position of the out-going packet within the out-going time-frame 35-3.
N

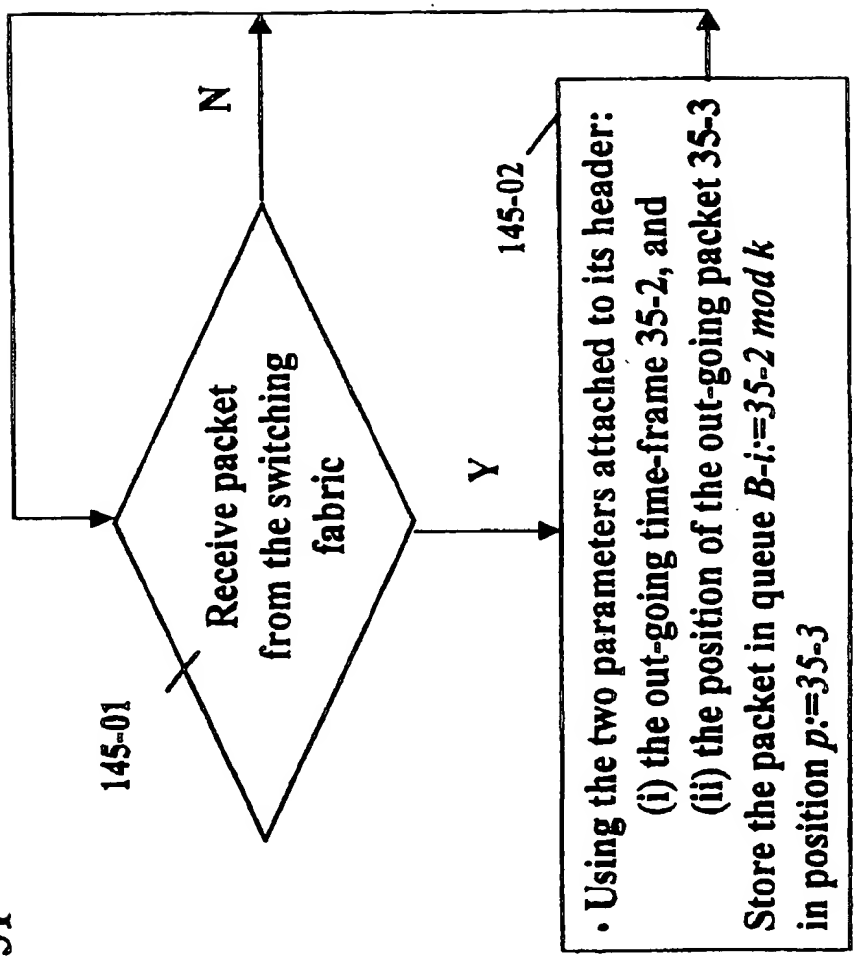
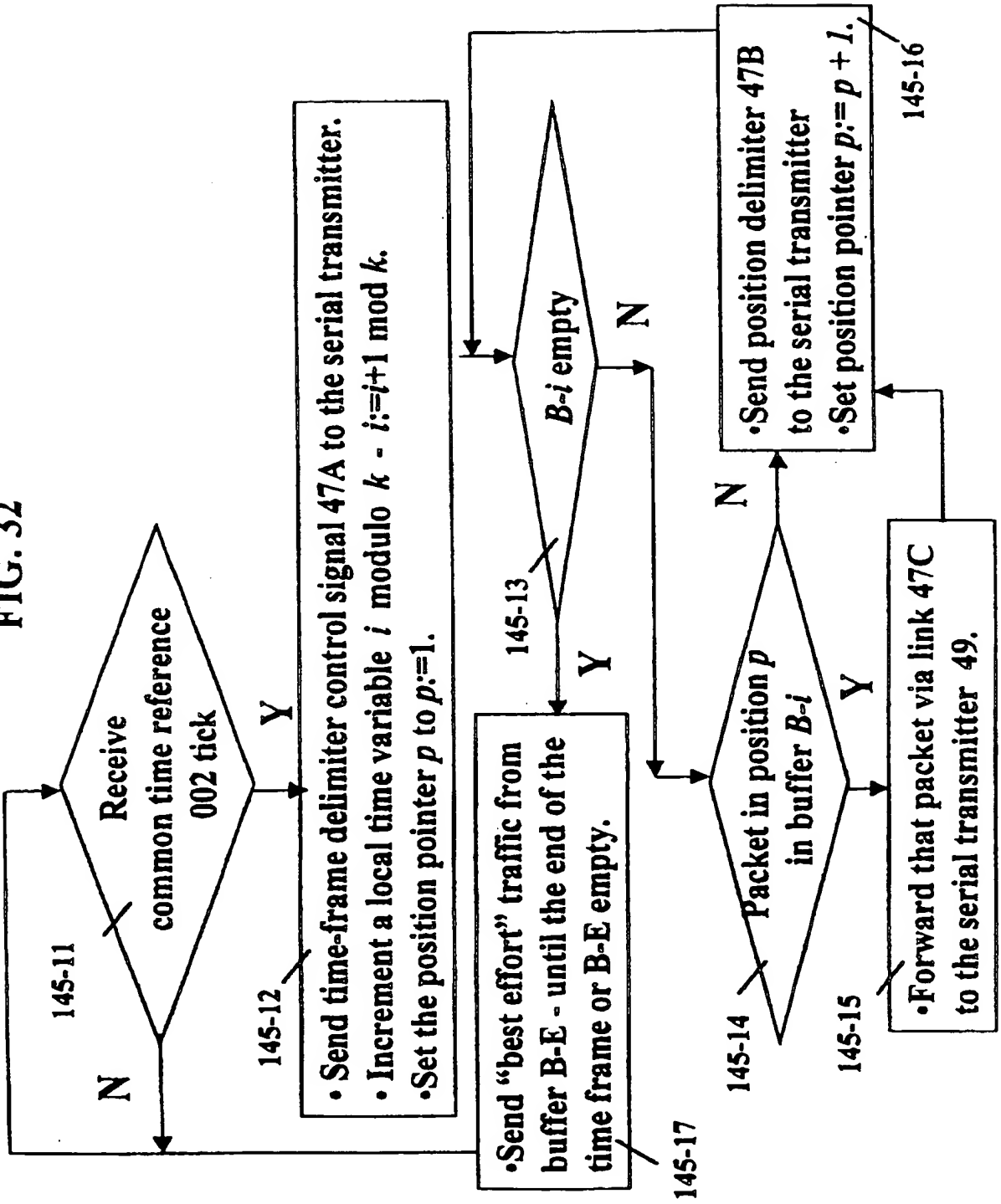

Receive packet from the switching fabric
145-01
N
Y
145-02
• Using the two parameters attached to its header:
(i) the out-going time-frame 35-2, and
(ii) the position of the out-going packet 35-3
Store the packet in queue B-i:=35-2 mod k
in position p:=35-3

FIG. 31

FIG. 32


145-11
Receive common time reference 002 tick
N
Y
145-12
• Send time-frame delimiter control signal 47A to the serial transmitter.
• Increment a local time variable i modulo k - i:=i+1 mod k.
•Set the position pointer p to p:=1.
145-13
B-i empty
N
Y
•Send "best effort" traffic from buffer B-E - until the end of the time frame or B-E empty.
145-17
145-14
Packet in position p in buffer B-i
N
Y
145-15
•Forward that packet via link 47C to the serial transmitter 49.
•Send position delimiter 47B to the serial transmitter
•Set position pointer p:= p + 1.
145-16

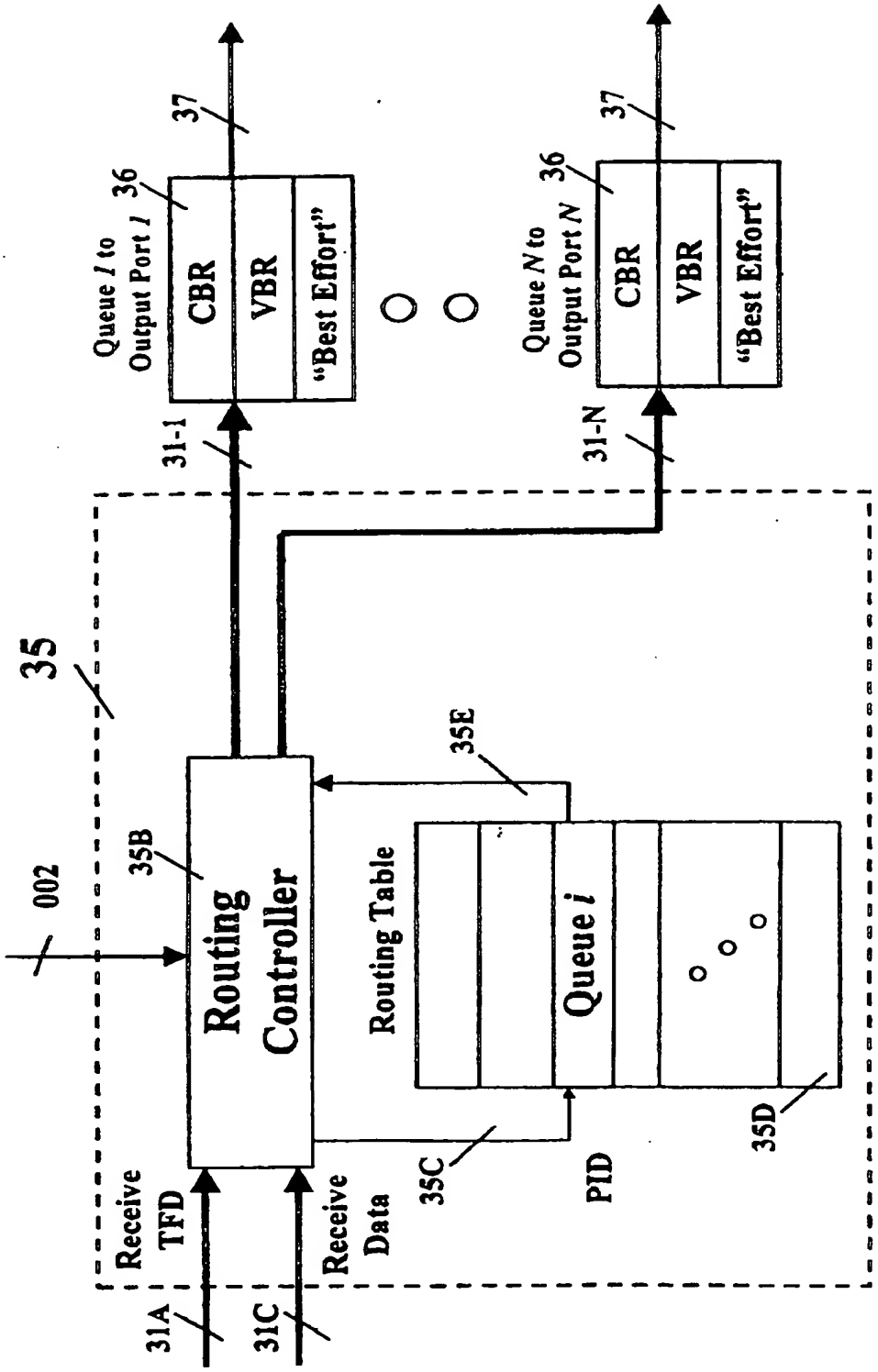

Queue 1 to Output Port 1
CBR
VBR
"Best Effort"
36
37
31-1
Queue N to Output Port N
CBR
VBR
"Best Effort"
36
37
31-N
35
002
35B
Routing Controller
35E
Routing Table
Queue i
35D
35C
PID
Receive TFD
Receive Data
31A
31C

FIG. 33

FIG. 34

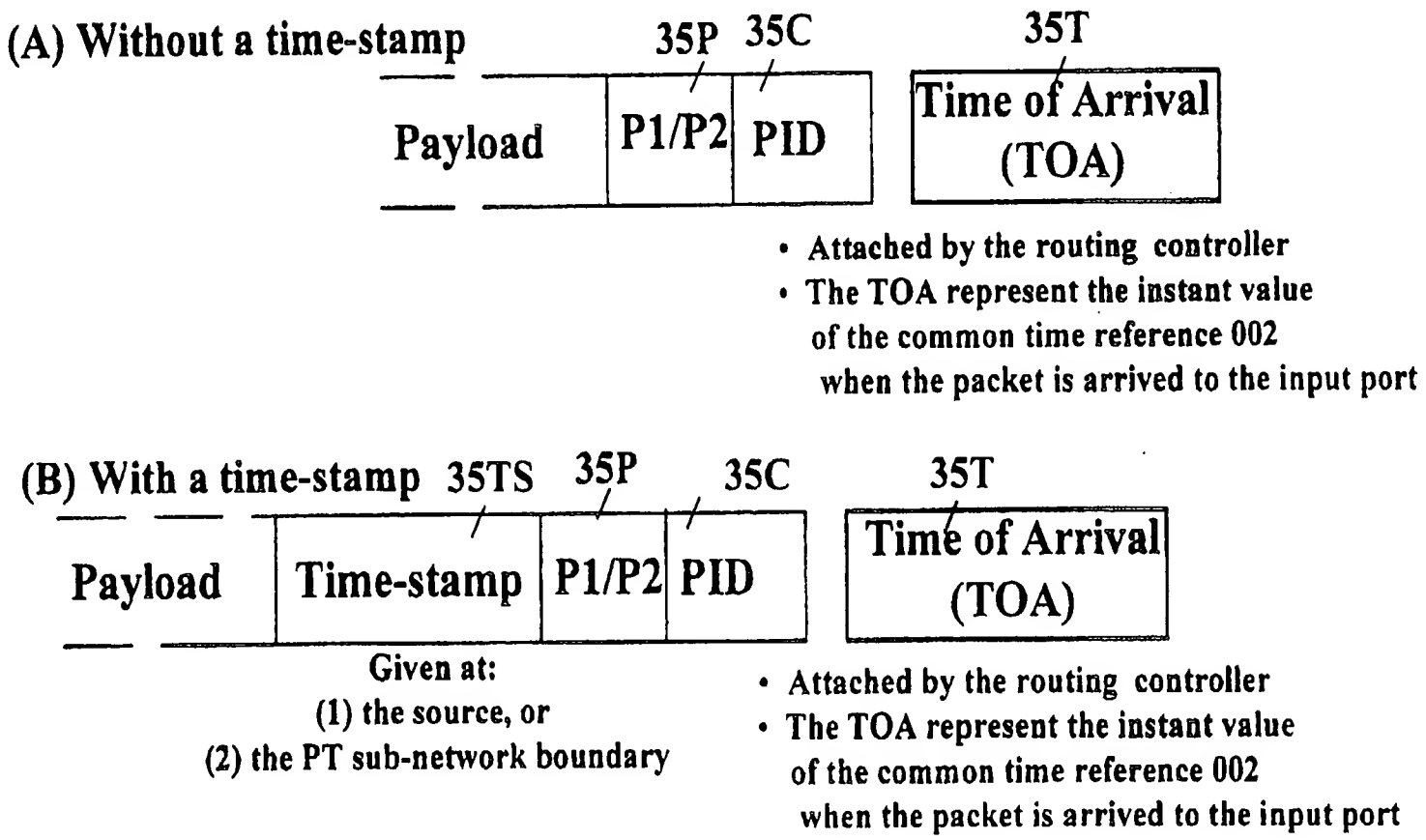

(A) Without a time-stamp
35P
35C
35T
Payload
P1/P2
PID
Time of Arrival (TOA)
• Attached by the routing controller
• The TOA represent the instant value of the common time reference 002 when the packet is arrived to the input port
(B) With a time-stamp
35TS
35P
35C
35T
Payload
Time-stamp
P1/P2
PID
Time of Arrival (TOA)
Given at:
(1) the source, or
(2) the PT sub-network boundary
• Attached by the routing controller
• The TOA represent the instant value of the common time reference 002 when the packet is arrived to the input port

FIG. 35

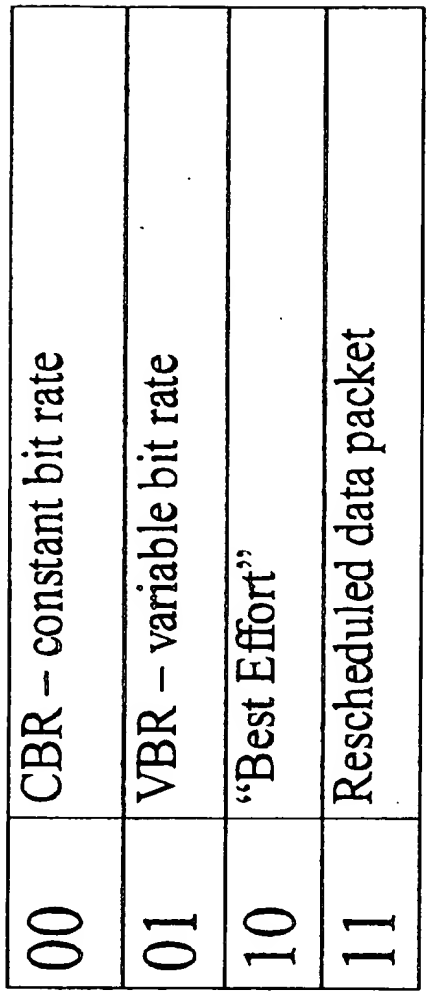
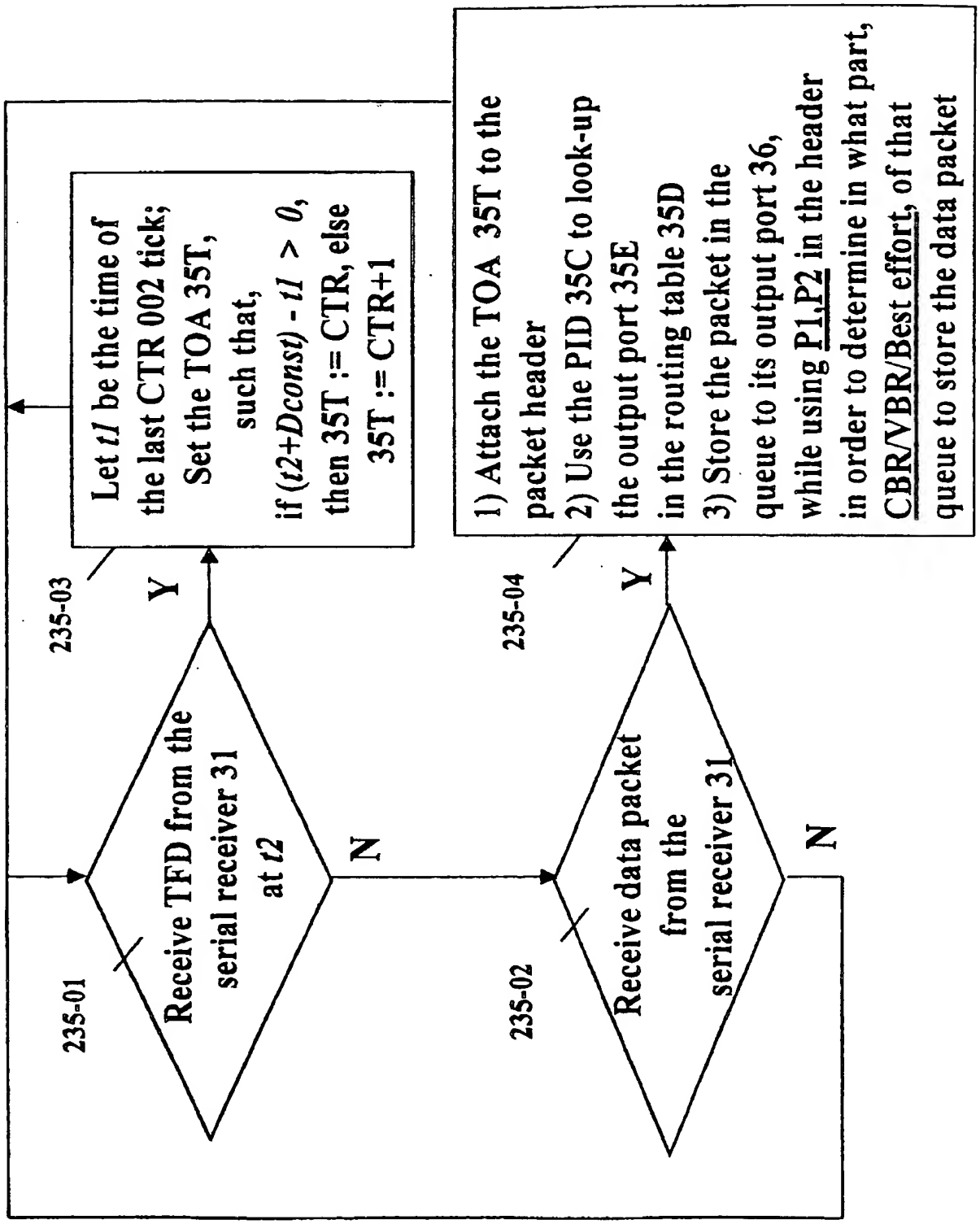

P1/P2
00
CBR – constant bit rate
01
VBR – variable bit rate
10
"Best Effort"
11
Rescheduled data packet

FIG. 36


235-01
Receive TFD from the serial receiver 31 at t2
Y
N
235-03
Let t1 be the time of the last CTR 002 tick; Set the TOA 35T, such that, if (t2+Dconst) - t1 > 0, then 35T := CTR, else 35T := CTR+1
235-02
Receive data packet from the serial receiver 31
Y
N
235-04
1) Attach the TOA 35T to the packet header
2) Use the PID 35C to look-up the output port 35E in the routing table 35D
3) Store the packet in the queue to its output port 36, while using P1,P2 in the header in order to determine in what part, CBR/VBR/Best effort, of that queue to store the data packet

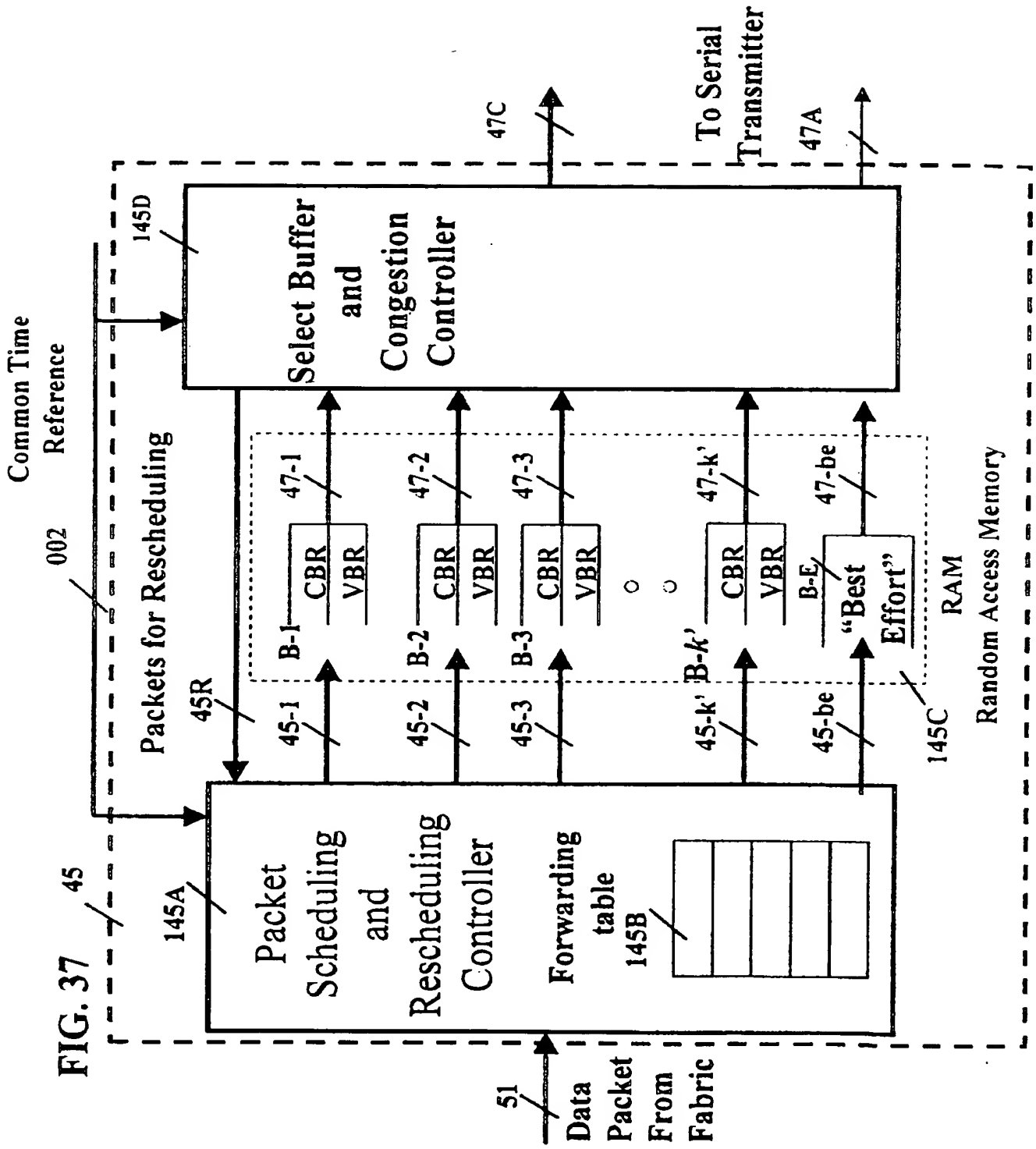

Common Time Reference
002
Packets for Rescheduling
45R
145D
Select Buffer and Congestion Controller
47C
To Serial Transmitter
47A
47-1
47-2
47-3
47-k'
47-be
B-1
B-2
B-3
B-k'
B-E
CBR
VBR
"Best Effort"
45-1
45-2
45-3
45-k'
45-be
145C
RAM
Random Access Memory
145A
Packet Scheduling and Rescheduling Controller
Forwarding table 145B
45
51
Data Packet From Fabric

FIG. 37

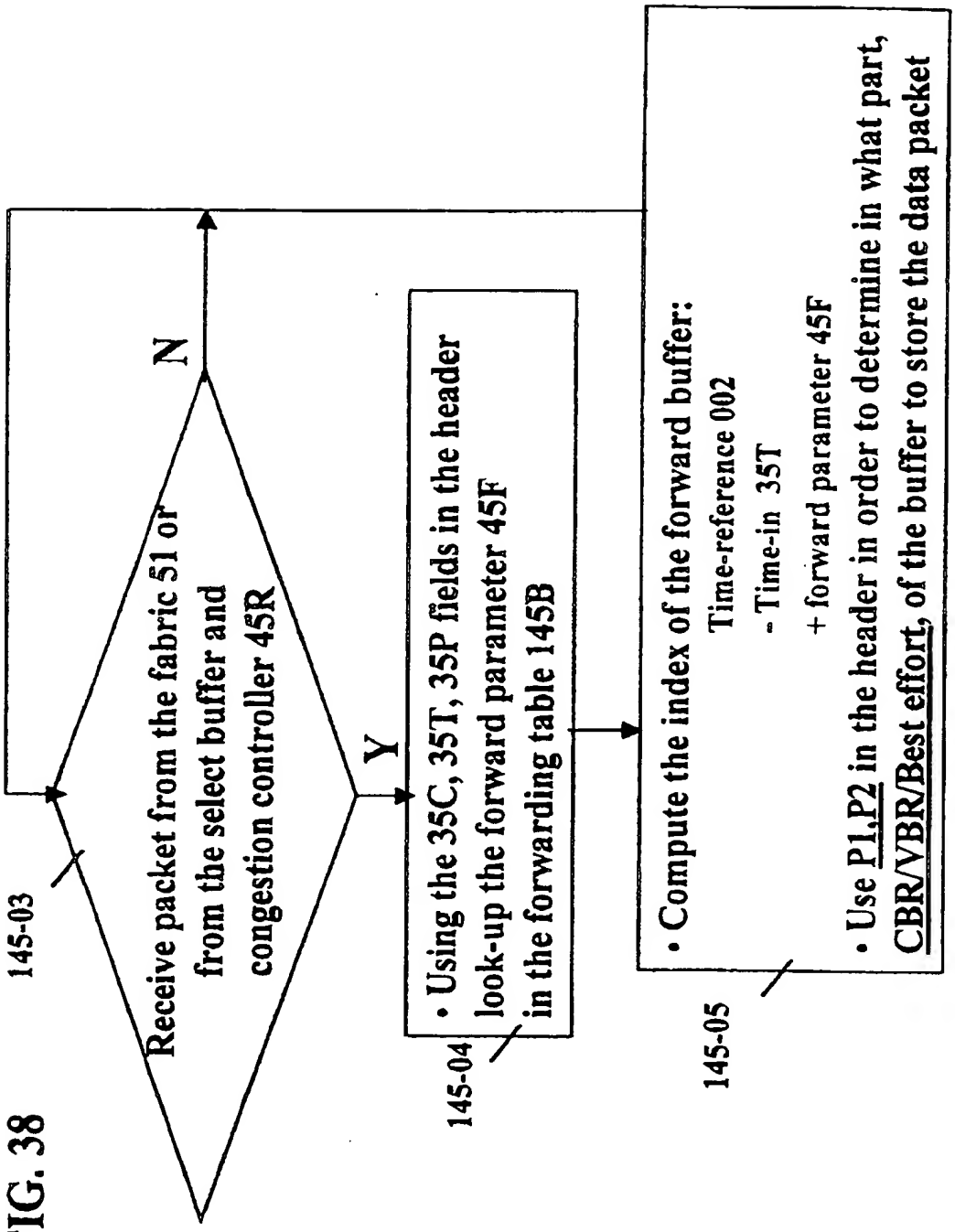
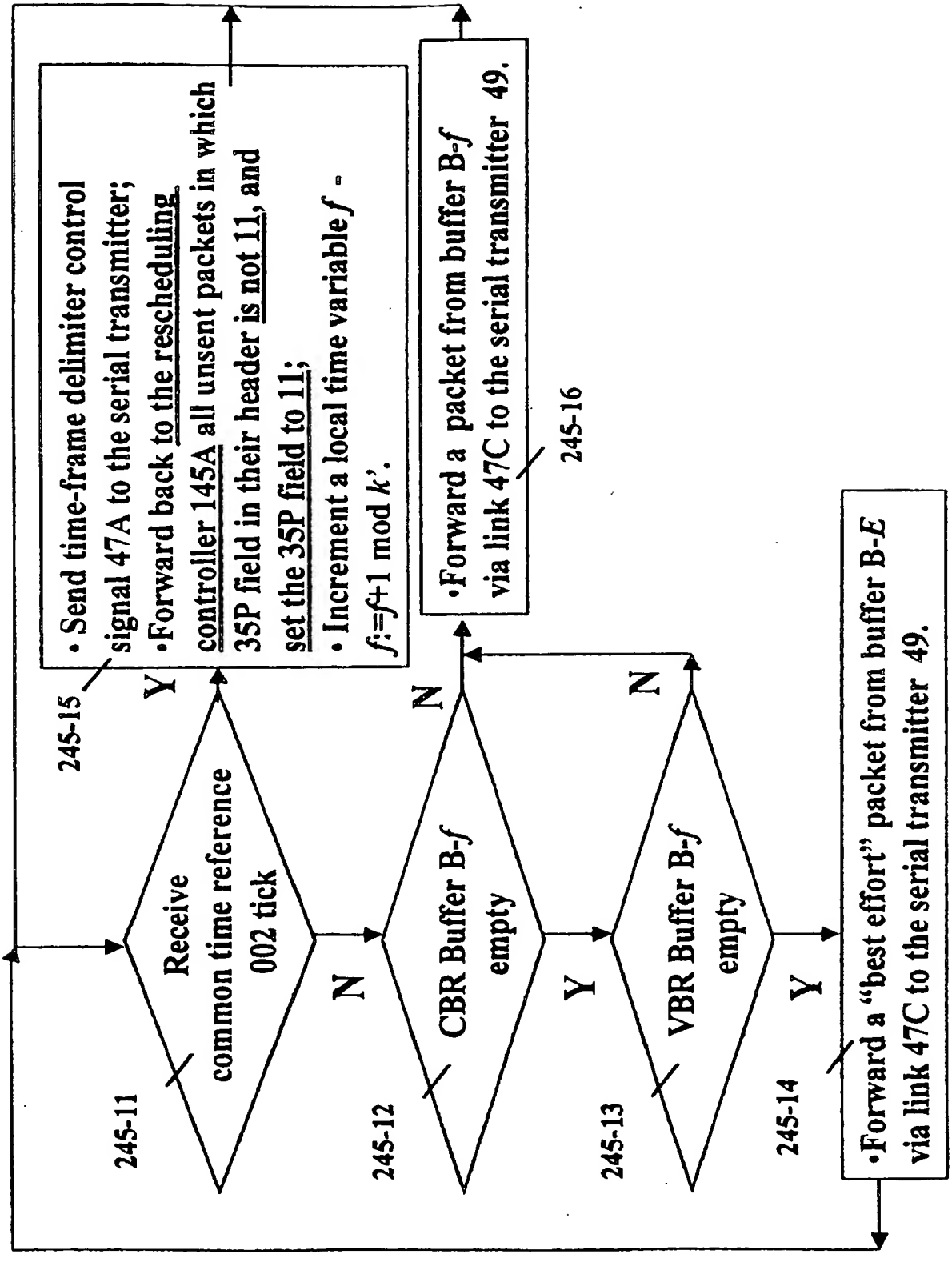

145-03
Receive packet from the fabric 51 or from the select buffer and congestion controller 45R
N
Y
145-04
• Using the 35C, 35T, 35P fields in the header look-up the forward parameter 45F in the forwarding table 145B
145-05
• Compute the index of the forward buffer:
Time-reference 002
- Time-in 35T
+ forward parameter 45F
• Use P1,P2 in the header in order to determine in what part, CBR/VBR/Best effort, of the buffer to store the data packet

FIG. 38

FIG. 39


245-11
Receive common time reference 002 tick
Y
245-15
• Send time-frame delimiter control signal 47A to the serial transmitter;
•Forward back to the rescheduling controller 145A all unsent packets in which 35P field in their header is not 11, and set the 35P field to 11;
• Increment a local time variable f = f:=f+1 mod k'.
N
245-12
CBR Buffer B-f empty
Y
245-16
•Forward a packet from buffer B-f via link 47C to the serial transmitter 49.
N
245-13
VBR Buffer B-f empty
Y
245-14
•Forward a "best effort" packet from buffer B-E via link 47C to the serial transmitter 49.

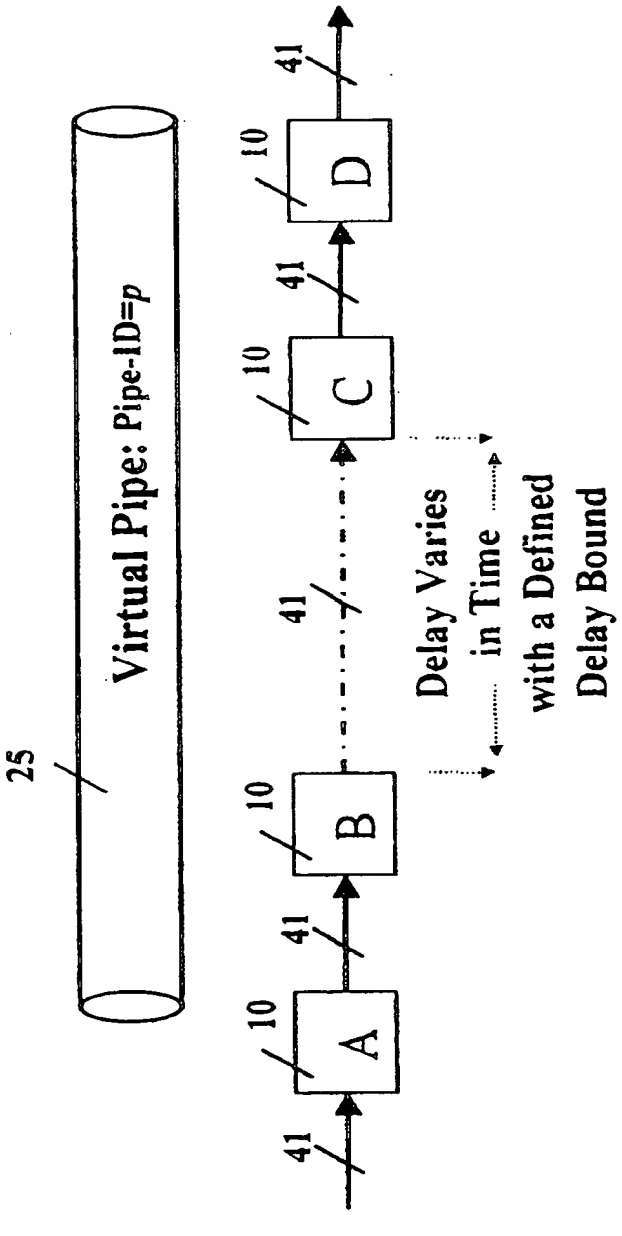

Virtual Pipe: Pipe-ID=p
25
A
B
C
D
10
41
Delay Varies
in Time
with a Defined
Delay Bound

FIG. 40

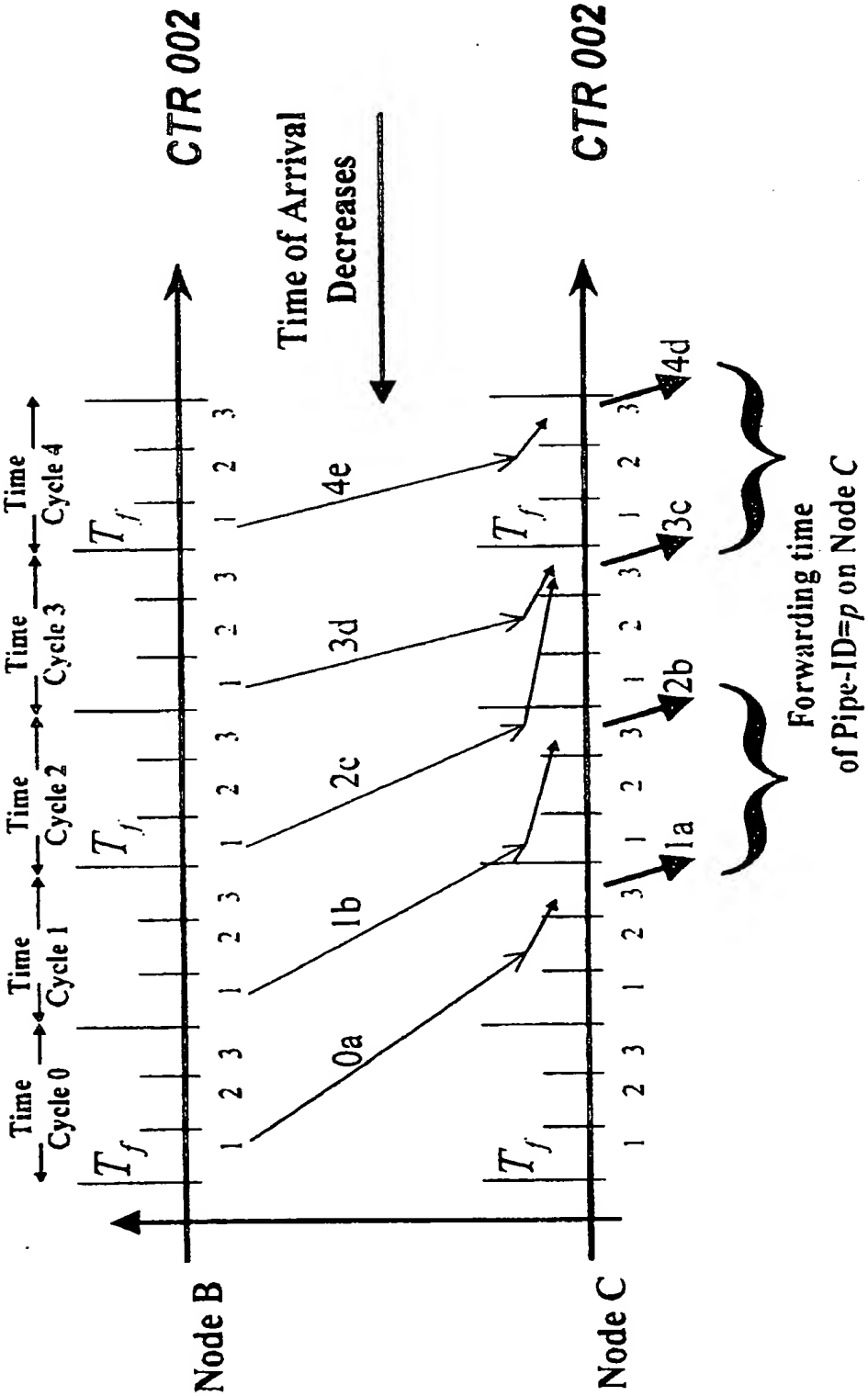

Time
Cycle 0
Time
Cycle 1
Time
Cycle 2
Time
Cycle 3
Time
Cycle 4
T_f
CTR 002
CTR 002
Time of Arrival
Decreases
0a
1b
2c
3d
4e
1a
2b
3c
4d
Node B
Node C
Forwarding time
of Pipe-ID=p on Node C

FIG. 41

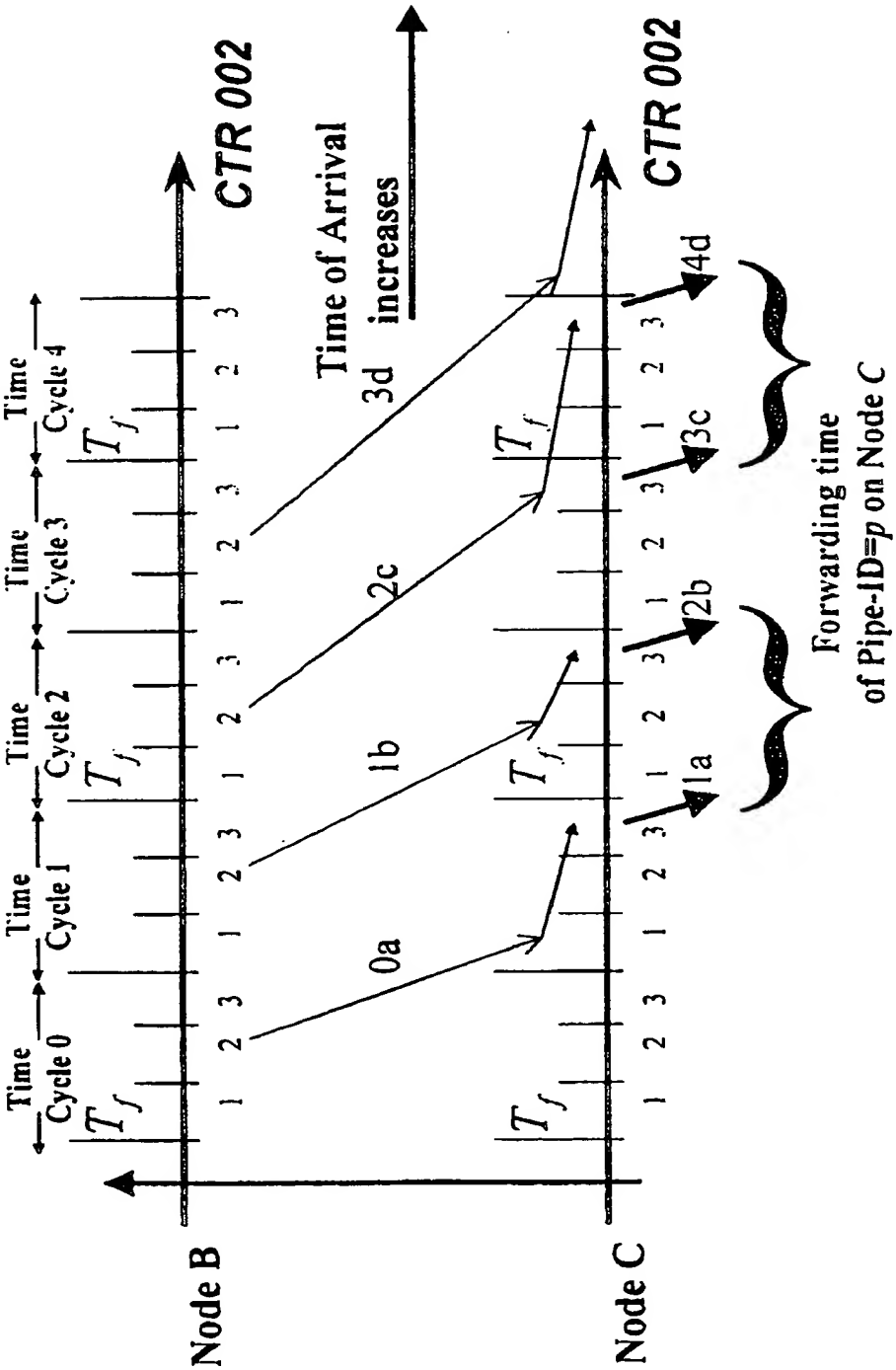

Time
Cycle 0
Time
Cycle 1
Time
Cycle 2
Time
Cycle 3
Time
Cycle 4
T_f
1 2 3
CTR 002
Node B
Node C
0a
1b
2c
3d
Time of Arrival increases
1a
2b
3c
4d
Forwarding time
of Pipe-ID=p on Node C

FIG. 42

FIG. 43

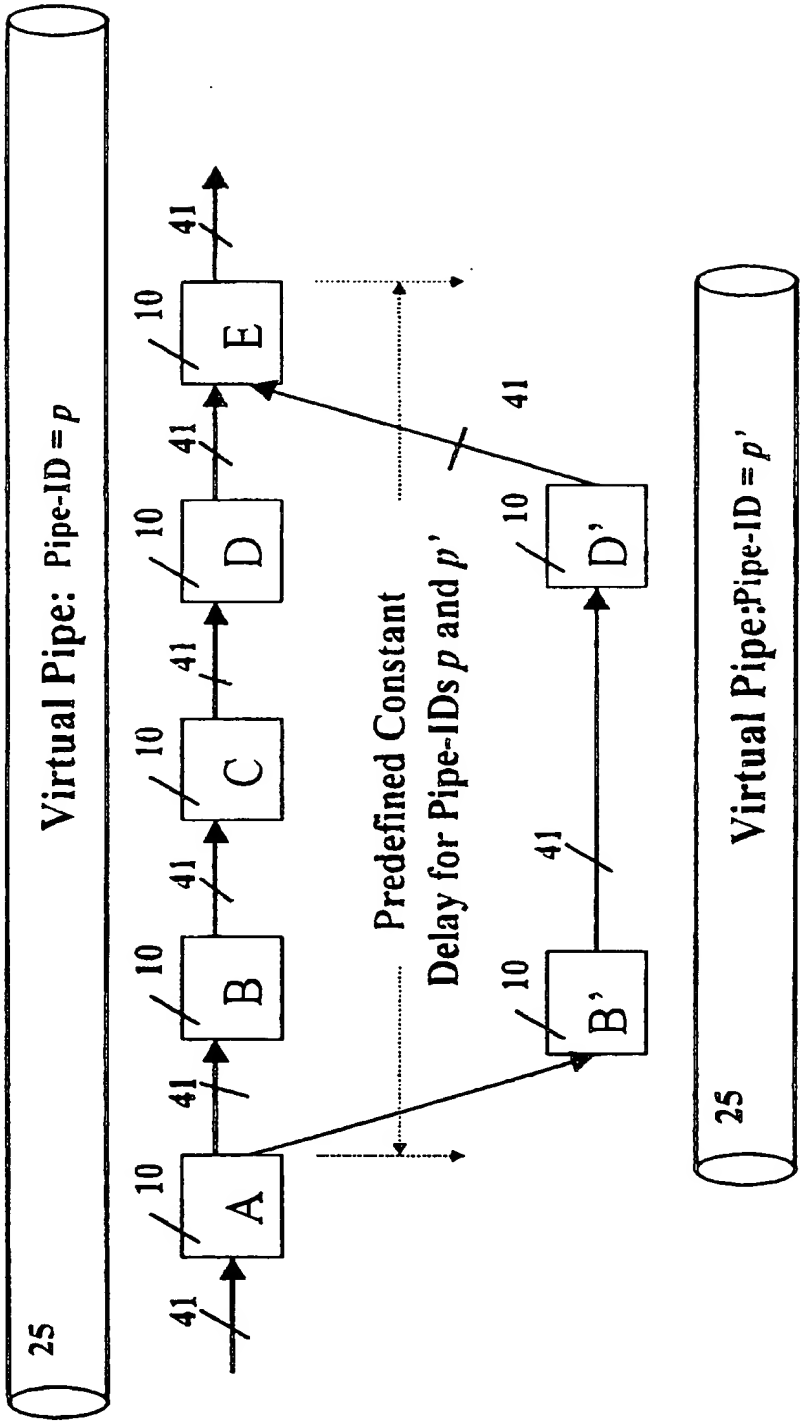
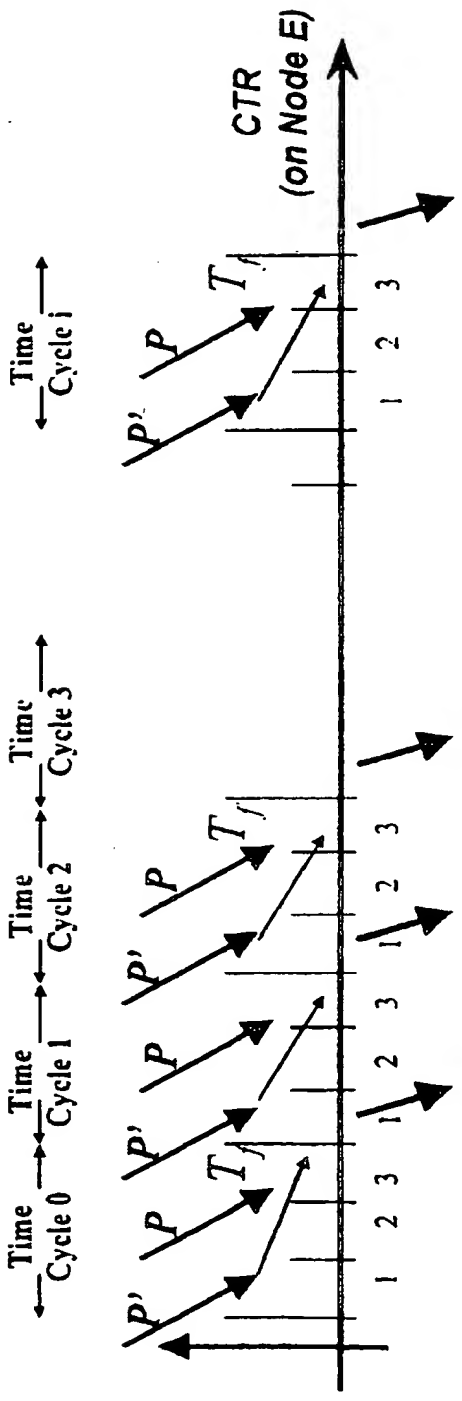

Virtual Pipe: Pipe-ID = p
25
A
B
C
D
E
B'
D'
10
41
Predefined Constant
Delay for Pipe-IDs p and p'
Virtual Pipe:Pipe-ID = p'
25

FIG. 44


Time
Cycle 0
Time
Cycle 1
Time
Cycle 2
Time
Cycle 3
Time
Cycle i
CTR
(on Node E)
P'
P
T_f
1 2 3
Forwarding time
out of Pipe-ID p and p' on Node E
and on Node A when data packets
from p'/p are forwarded to p/p'

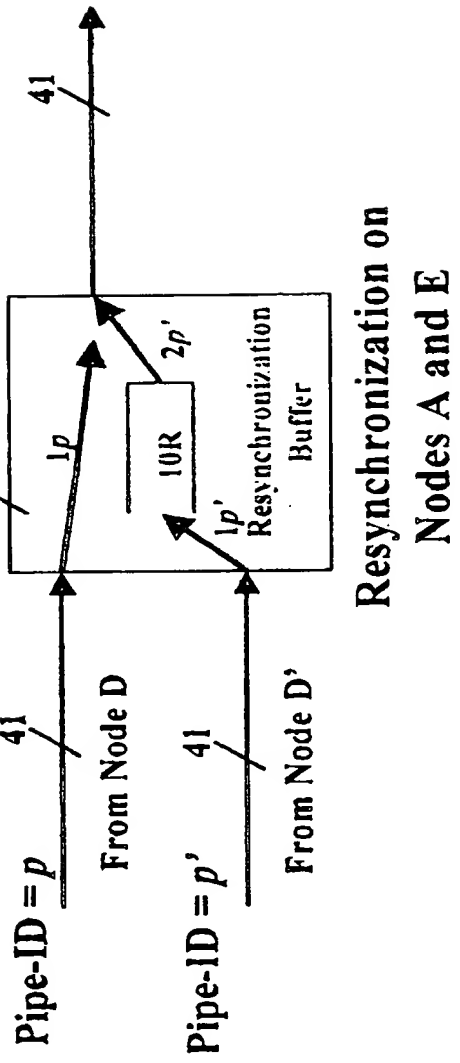

Pipe-ID = p
41
From Node D
Pipe-ID = p'
41
From Node D'
10
1p
10R
1p'
Resynchronization Buffer
2p'
41
Resynchronization on Nodes A and E

FIG. 45

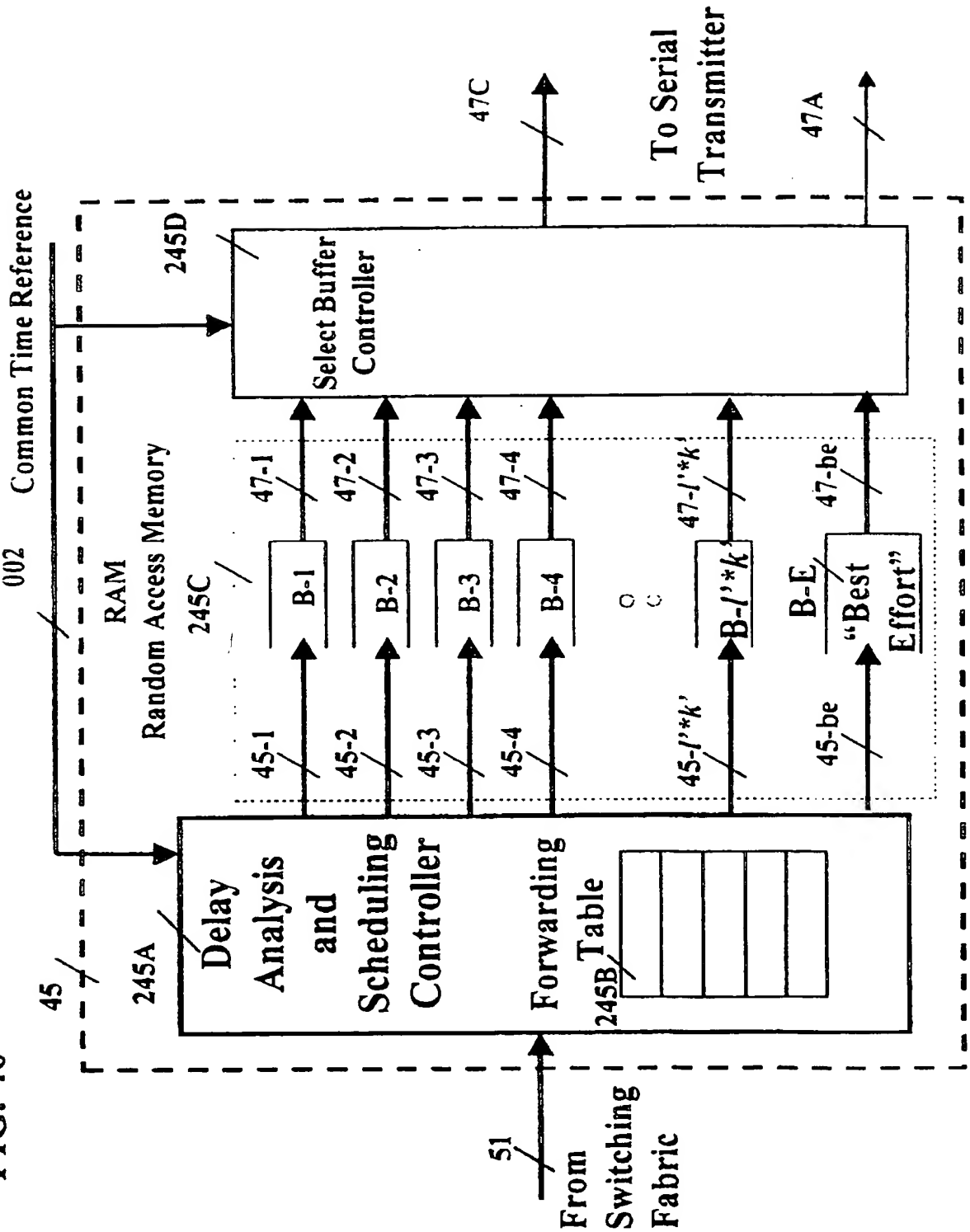
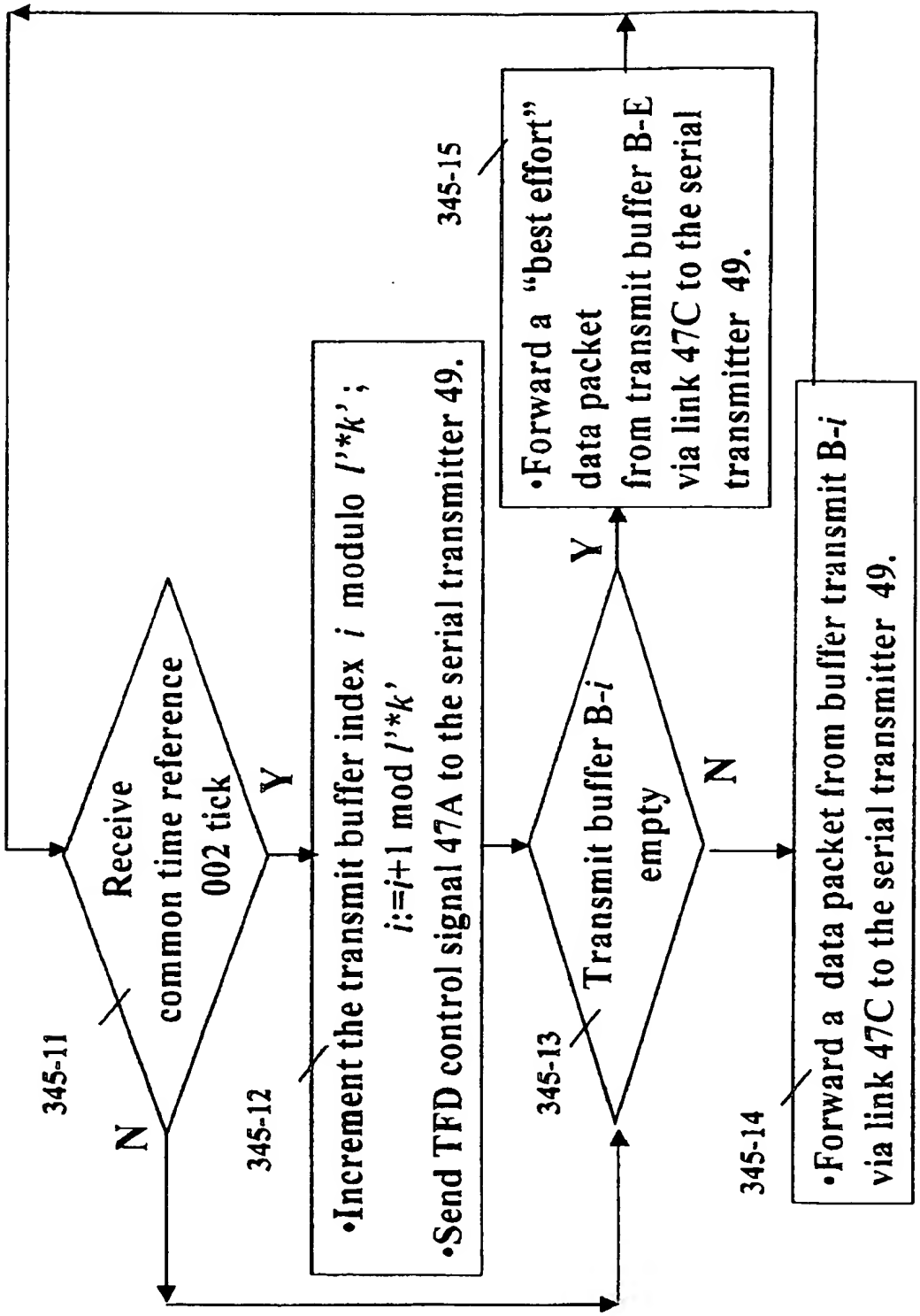

45
245A
Delay Analysis and Scheduling Controller
Forwarding Table
245B
002
Common Time Reference
RAM
Random Access Memory
245C
245D
Select Buffer Controller
45-1
45-2
45-3
45-4
45-l''*k'
45-be
47-1
47-2
47-3
47-4
47-l''*k'
47-be
B-1
B-2
B-3
B-4
B-l''*k'
B-E
"Best Effort"
47C
47A
To Serial Transmitter
51
From Switching Fabric

FIG. 46

FIG. 47


345-11
Receive common time reference 002 tick
N
Y
345-12
•Increment the transmit buffer index i modulo l'*k'; i:=i+1 mod l'*k'
•Send TFD control signal 47A to the serial transmitter 49.
345-13
Transmit buffer B-i empty
Y
N
345-15
•Forward a "best effort" data packet from transmit buffer B-E via link 47C to the serial transmitter 49.
345-14
•Forward a data packet from buffer transmit B-i via link 47C to the serial transmitter 49.

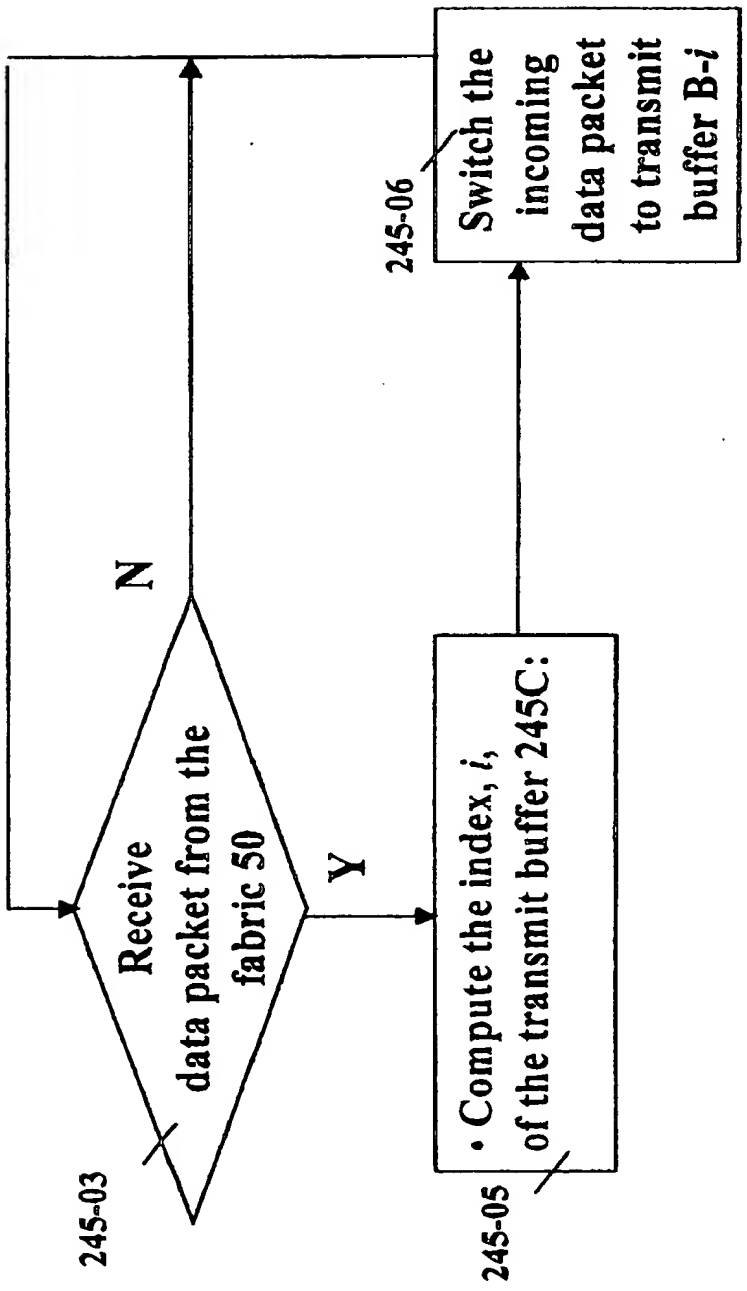

Receive data packet from the fabric 50
245-03
N
Y
• Compute the index, i, of the transmit buffer 245C:
245-05
Switch the incoming data packet to transmit buffer B-i
245-06

FIG. 48

FIG. 49

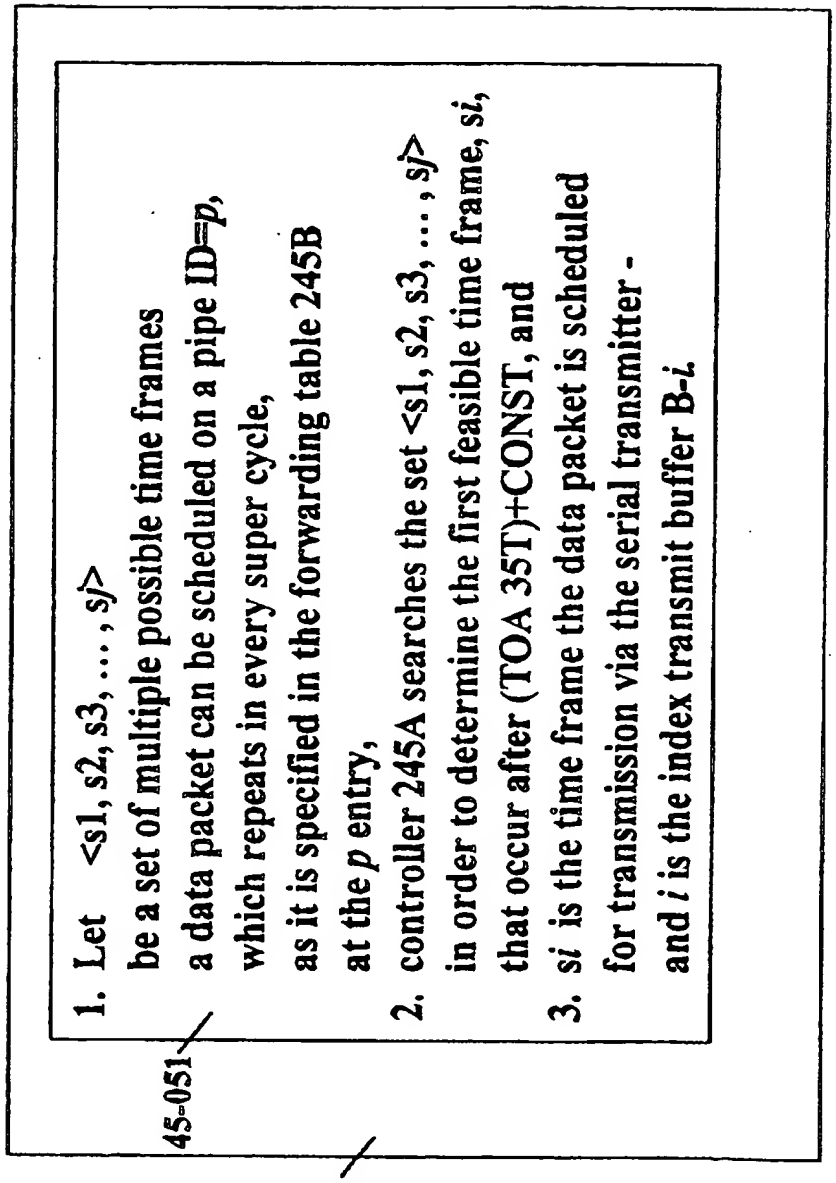

1. Let <s1, s2, s3, ... , sj>
be a set of multiple possible time frames
a data packet can be scheduled on a pipe ID=p,
which repeats in every super cycle,
as it is specified in the forwarding table 245B
at the p entry;
2. controller 245A searches the set <s1, s2, s3, ... , sj>
in order to determine the first feasible time frame, si,
that occur after (TOA 35T)+CONST, and
3. si is the time frame the data packet is scheduled
for transmission via the serial transmitter -
and i is the index transmit buffer B-i.
45-051
45-05

FIG. 50

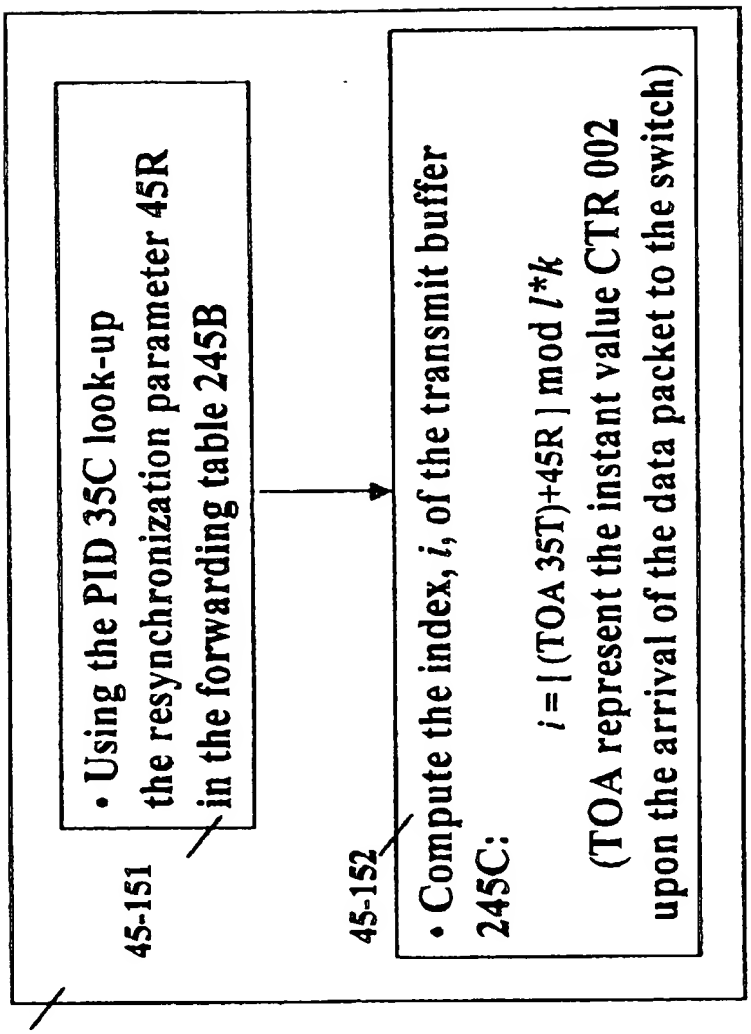

45-05
45-151
• Using the PID 35C look-up the resynchronization parameter 45R in the forwarding table 245B
45-152
• Compute the index, i, of the transmit buffer 245C:
i = ⌊(TOA 35T)+45R⌋ mod l*k
(TOA represent the instant value CTR 002 upon the arrival of the data packet to the switch)

FIG. 51

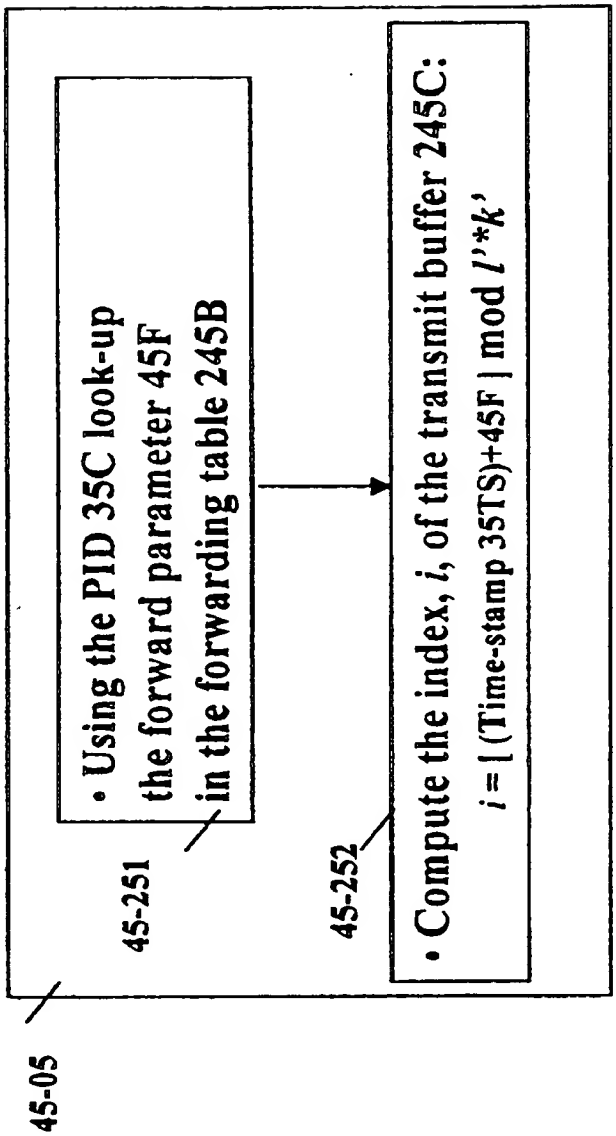
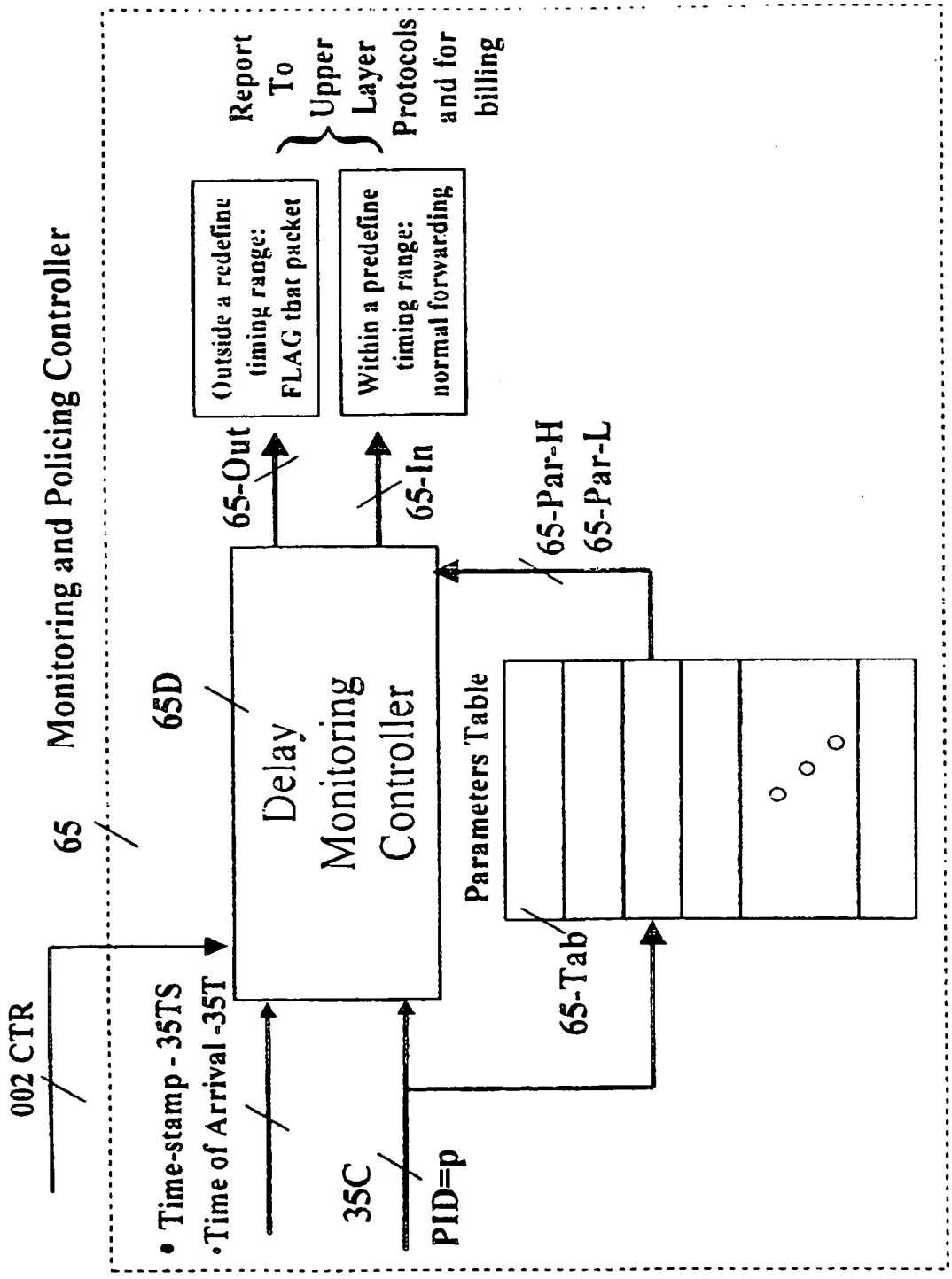

45-05
45-251
• Using the PID 35C look-up the forward parameter 45F in the forwarding table 245B
45-252
• Compute the index, i, of the transmit buffer 245C:
i = | (Time-stamp 35TS)+45F | mod l'*k'

FIG. 52


002 CTR
65
Monitoring and Policing Controller
• Time-stamp - 35TS
•Time of Arrival -35T
35C
PID=p
65D
Delay
Monitoring
Controller
65-Out
65-In
65-Par-H
65-Par-L
65-Tab
Parameters Table
Outside a redefine timing range: FLAG that packet
Within a predefine timing range: normal forwarding
Report To Upper Layer Protocols and for billing

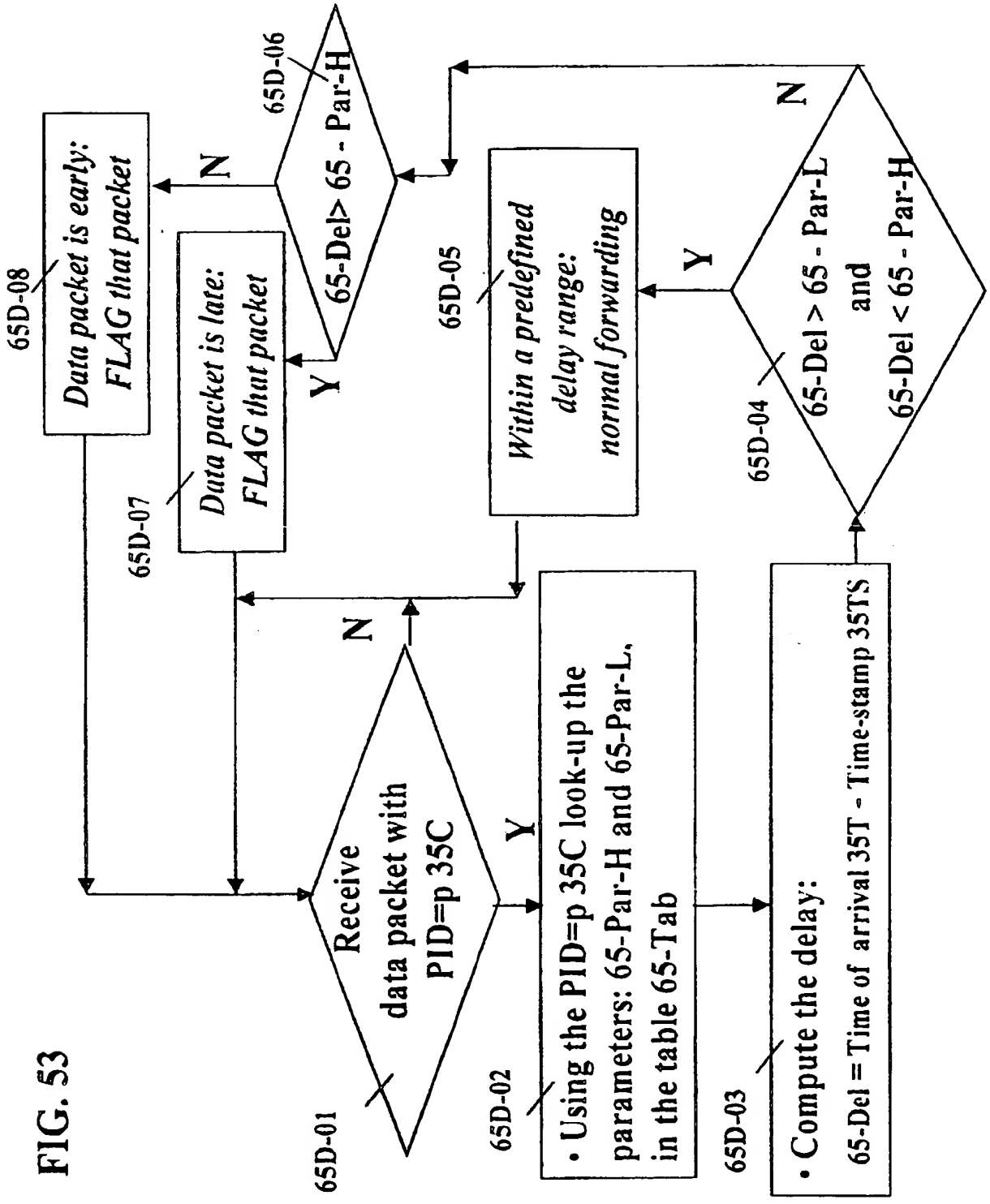
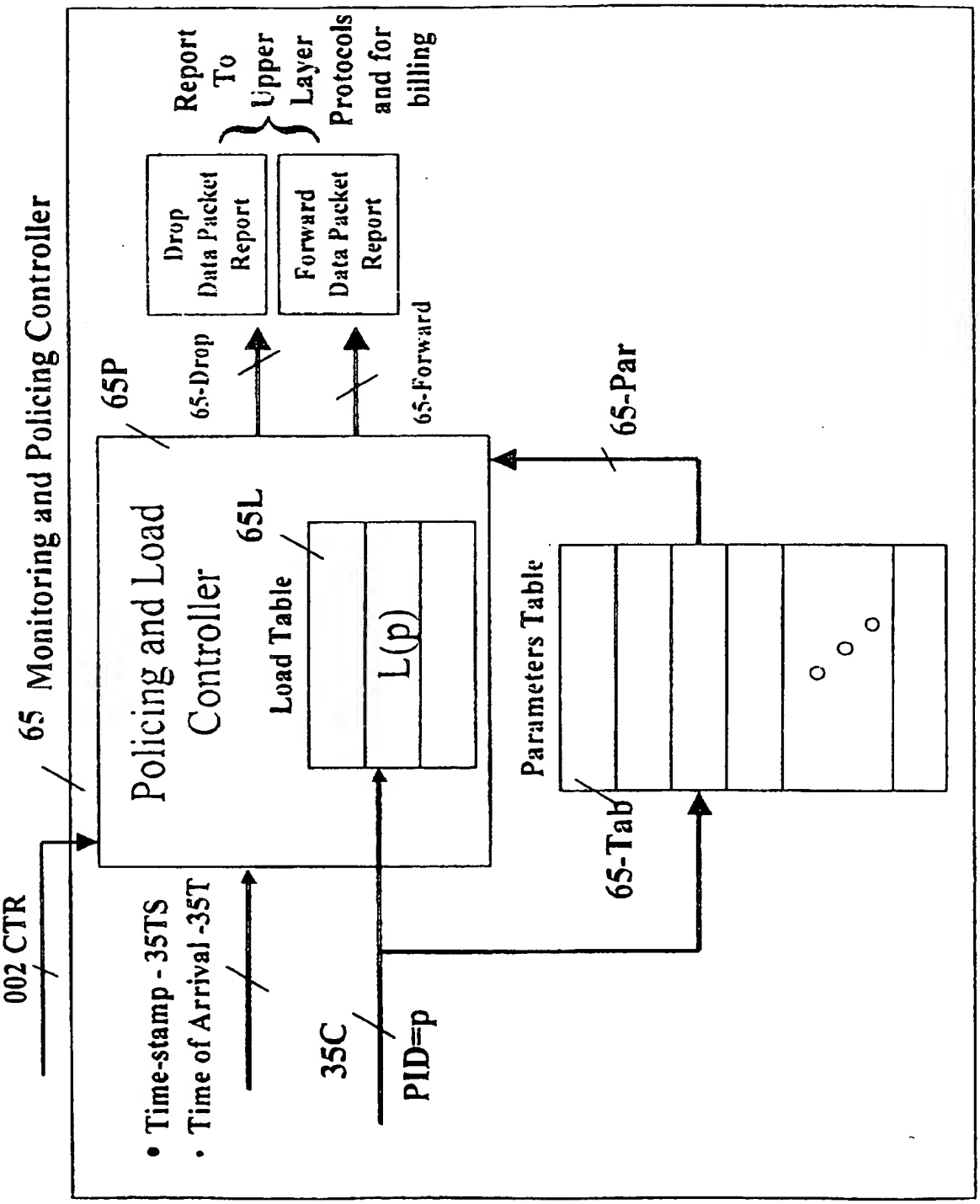

Receive data packet with PID=p 35C
65D-01
N
Y
• Using the PID=p 35C look-up the parameters: 65-Par-H and 65-Par-L, in the table 65-Tab
65D-02
• Compute the delay:
65-Del = Time of arrival 35T - Time-stamp 35TS
65D-03
65-Del > 65 - Par-L and 65-Del < 65 - Par-H
65D-04
N
Y
Within a predefined delay range: normal forwarding
65D-05
65-Del> 65 - Par-H
65D-06
N
Y
Data packet is late: FLAG that packet
65D-07
Data packet is early: FLAG that packet
65D-08

FIG. 53

FIG. 54


65 Monitoring and Policing Controller
002 CTR
• Time-stamp - 35TS
• Time of Arrival -35T
35C
PID=p
Policing and Load Controller
65P
65L
Load Table
L(p)
Parameters Table
65-Tab
65-Par
65-Drop
65-Forward
Drop Data Packet Report
Forward Data Packet Report
Report To Upper Layer Protocols and for billing

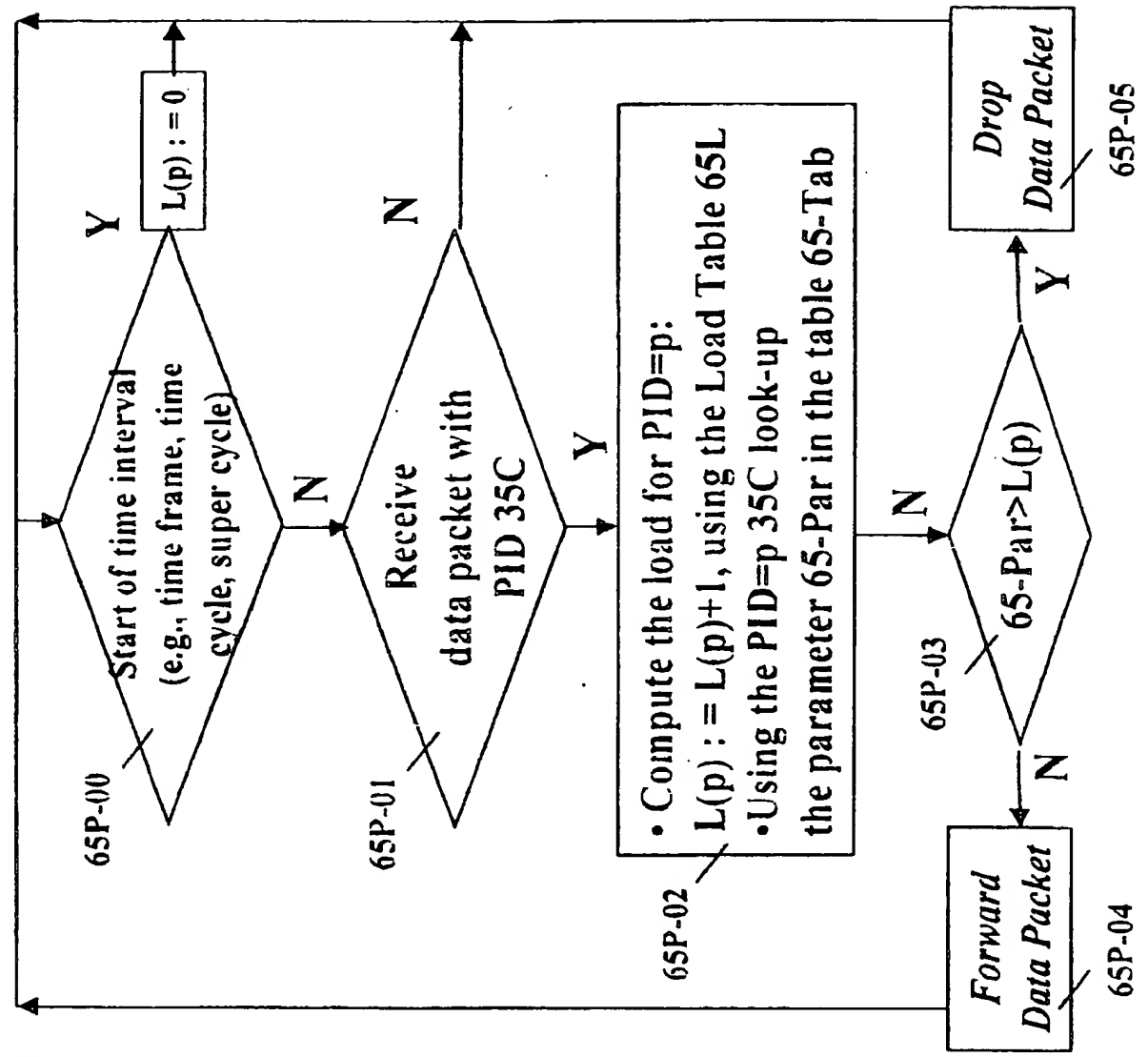

Start of time interval (e.g., time frame, time cycle, super cycle)
65P-00
Y
L(p) : = 0
N
Receive data packet with PID 35C
65P-01
N
Y
• Compute the load for PID=p: L(p) : = L(p)+1, using the Load Table 65L
• Using the PID=p 35C look-up the parameter 65-Par in the table 65-Tab
65P-02
N
65-Par>L(p)
65P-03
Y
Drop Data Packet
65P-05
N
Forward Data Packet
65P-04

FIG. 55

FIG. 56

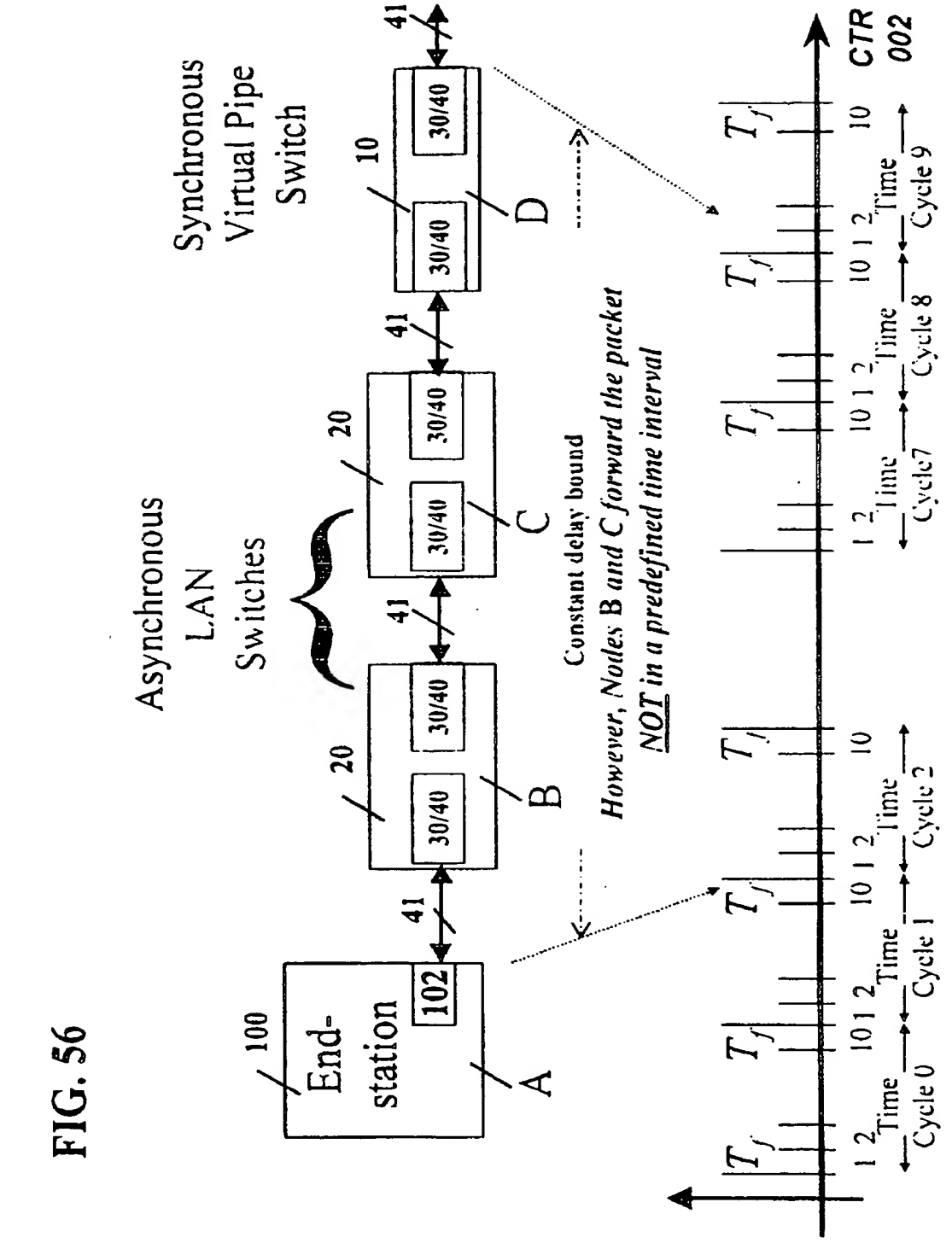

Asynchronous LAN Switches
Synchronous Virtual Pipe Switch
End-station
100
102
A
B
C
D
20
10
41
30/40
Constant delay bound
However, Nodes B and C forward the packet NOT in a predefined time interval
T_f
Time
Cycle 0
Cycle 1
Cycle 2
Cycle7
Cycle 8
Cycle 9
CTR 002

FIG. 57

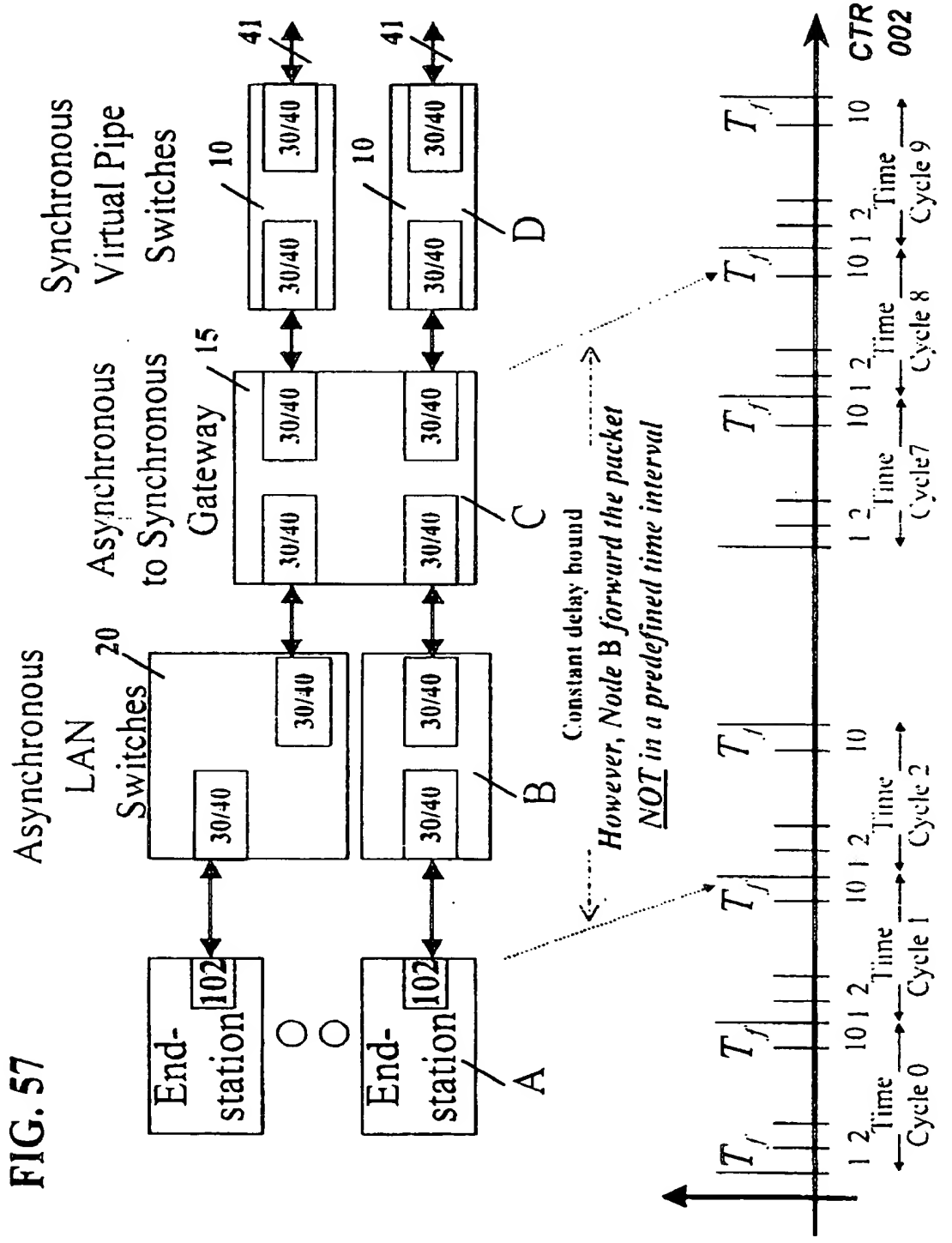

Asynchronous LAN Switches
Asynchronous to Synchronous Gateway
Synchronous Virtual Pipe Switches
End-station
102
30/40
20
15
10
41
A
B
C
D
Constant delay bound
However, Node B forward the packet
NOT in a predefined time interval
Time
Cycle 0
Cycle 1
Cycle 2
Cycle7
Cycle 8
Cycle 9
CTR
002

FIG. 58

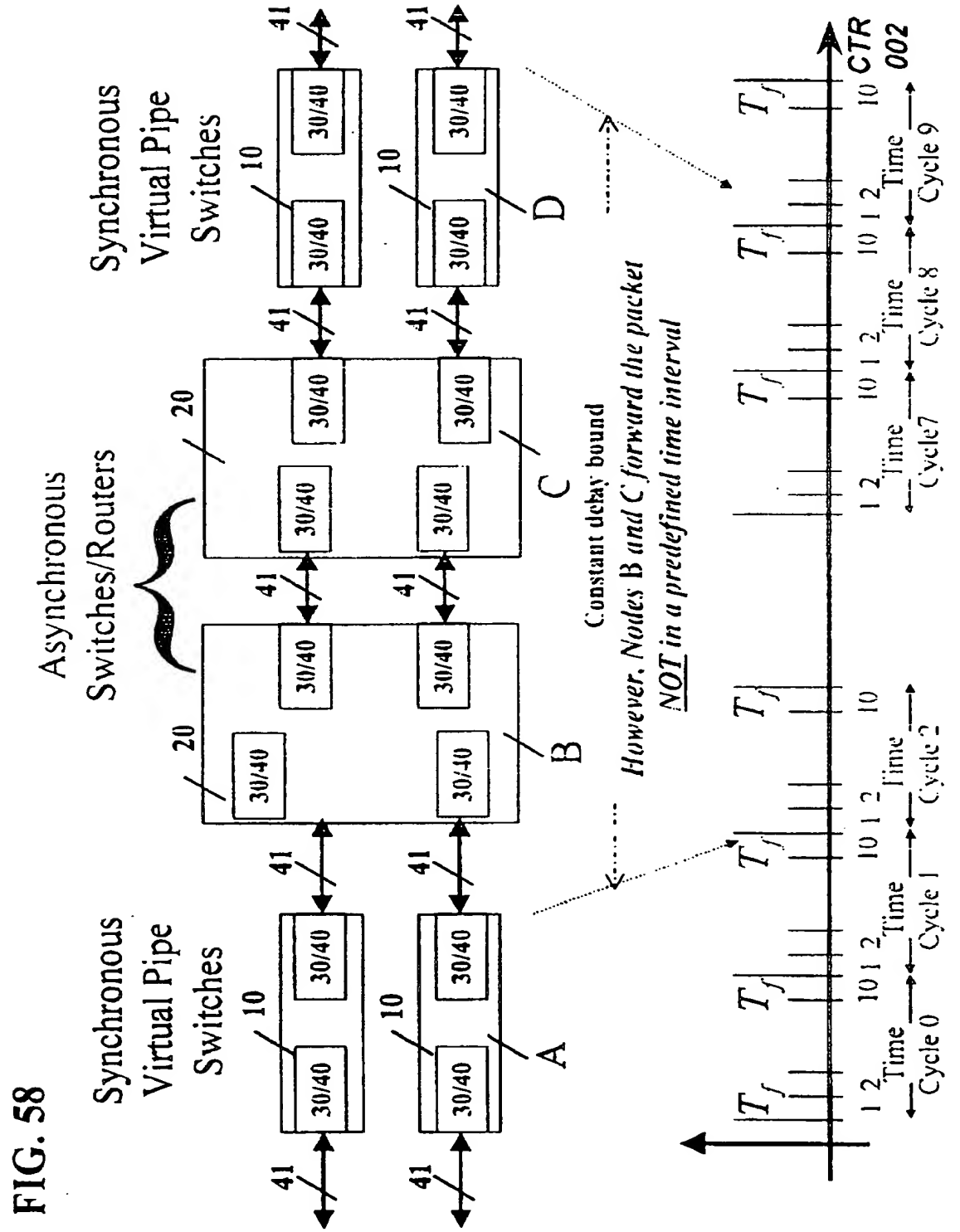

Synchronous Virtual Pipe Switches
Asynchronous Switches/Routers
Synchronous Virtual Pipe Switches
30/40
41
10
20
A
B
C
D
Constant delay bound
However, Nodes B and C forward the packet
NOT in a predefined time interval
T_f
Time
Cycle 0
Cycle 1
Cycle 2
Cycle7
Cycle 8
Cycle 9
CTR
002

FIG. 59

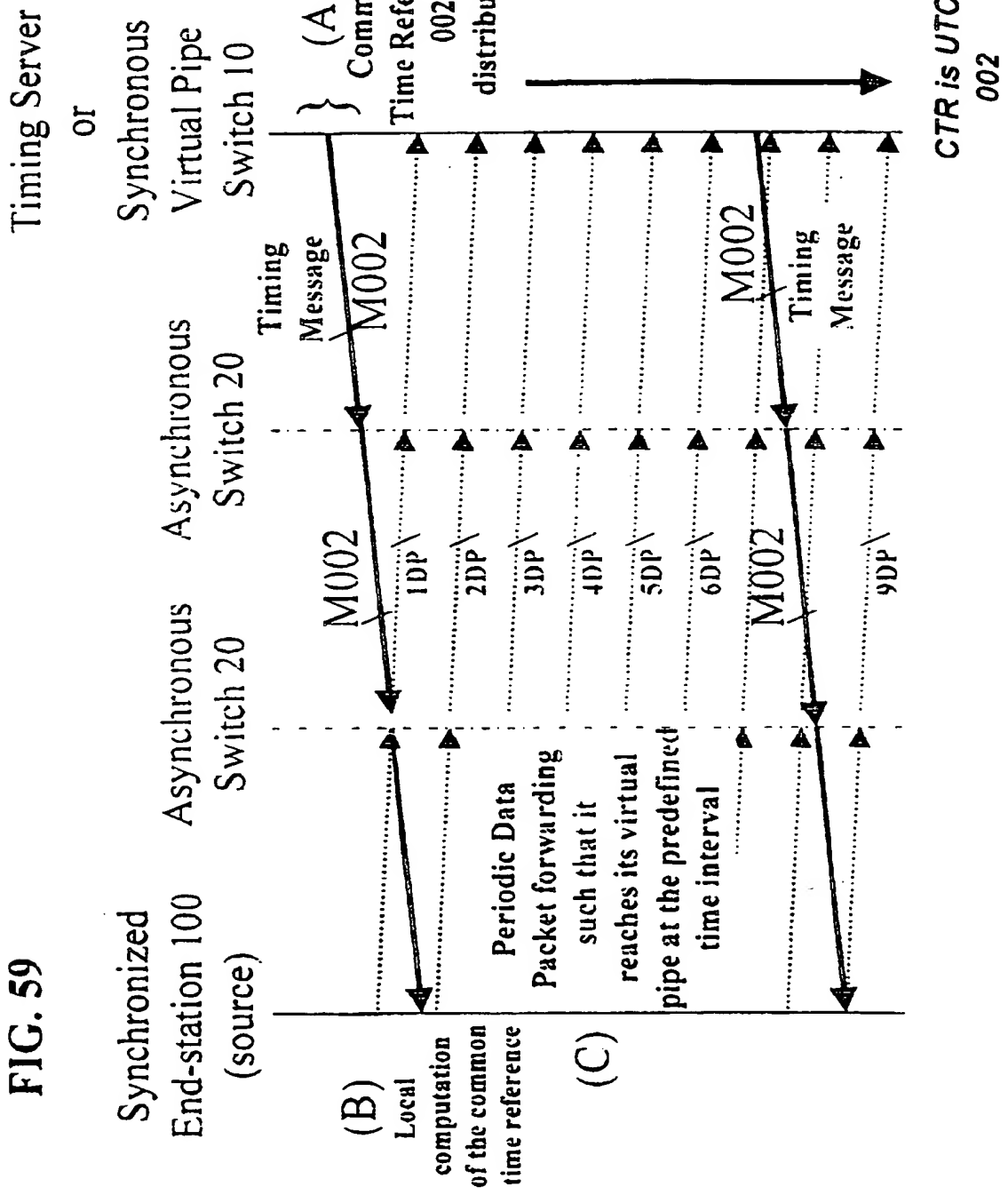

Timing Server
or
Synchronous
Virtual Pipe
Switch 10
(A)
Common
Time Reference
002
distribution
CTR is UTC
002
Synchronized
End-station 100
(source)
Asynchronous
Switch 20
Asynchronous
Switch 20
Timing
Message
M002
M002
M002
M002
Timing
Message
1DP
2DP
3DP
4DP
5DP
6DP
9DP
(B)
Local
computation
of the common
time reference
(C)
Periodic Data
Packet forwarding
such that it
reaches its virtual
pipe at the predefined
time interval

FIG. 60

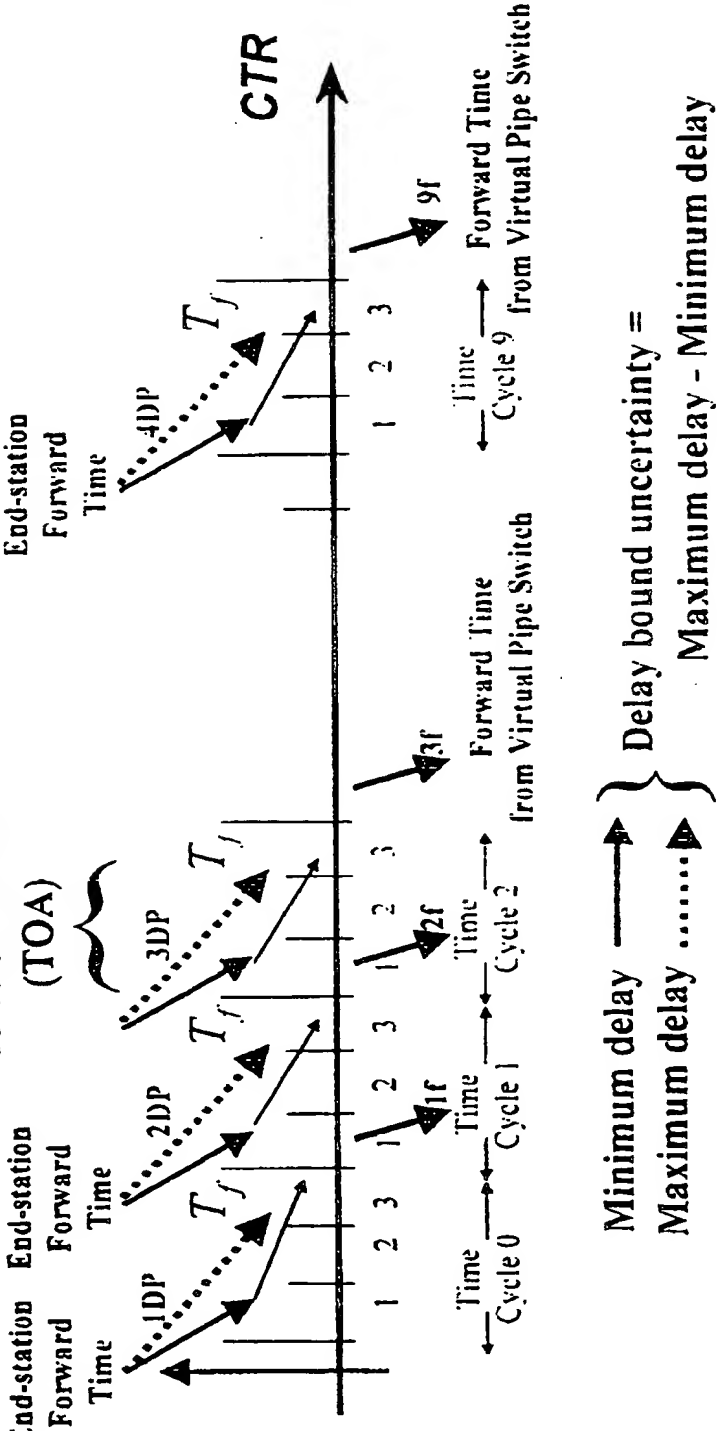

Possible
Time of Arrival
(TOA)
End-station
Forward
Time
1DP
2DP
3DP
4DP
T_f
CTR
1 2 3
1f
2f
3f
9f
Time
Cycle 0
Cycle 1
Cycle 2
Cycle 9
Forward Time
from Virtual Pipe Switch
Minimum delay
Maximum delay
Delay bound uncertainty =
Maximum delay - Minimum delay

FIG. 61

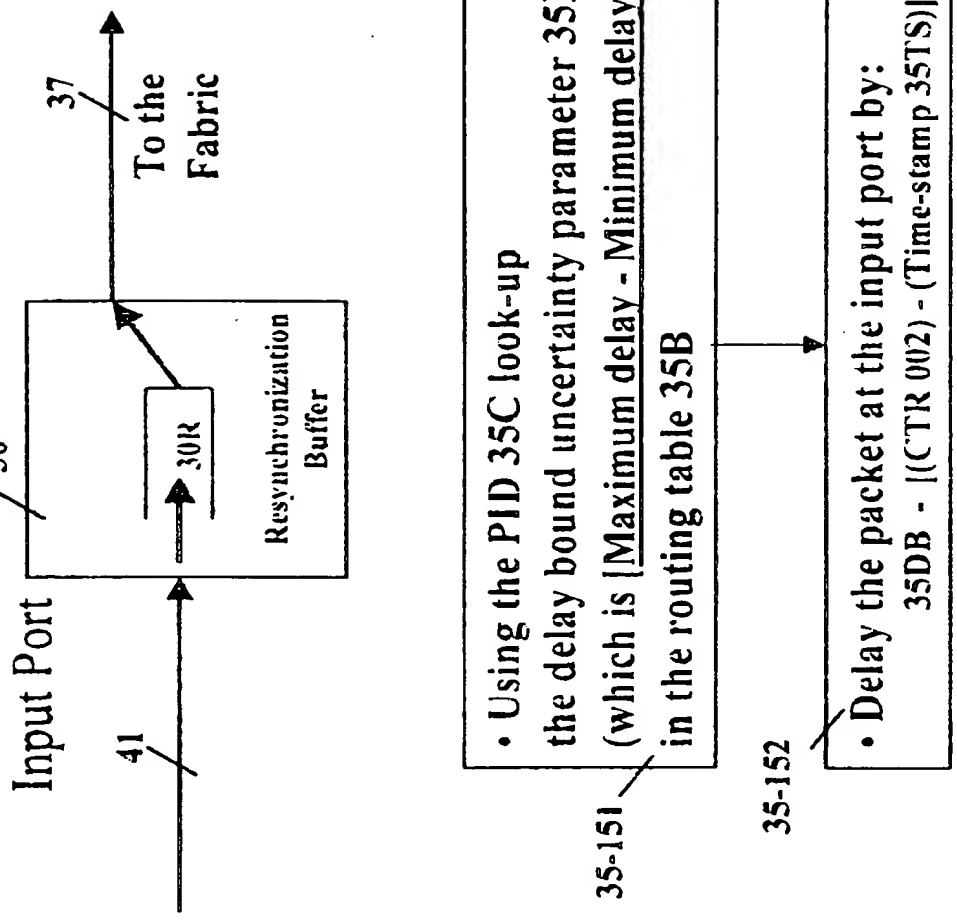
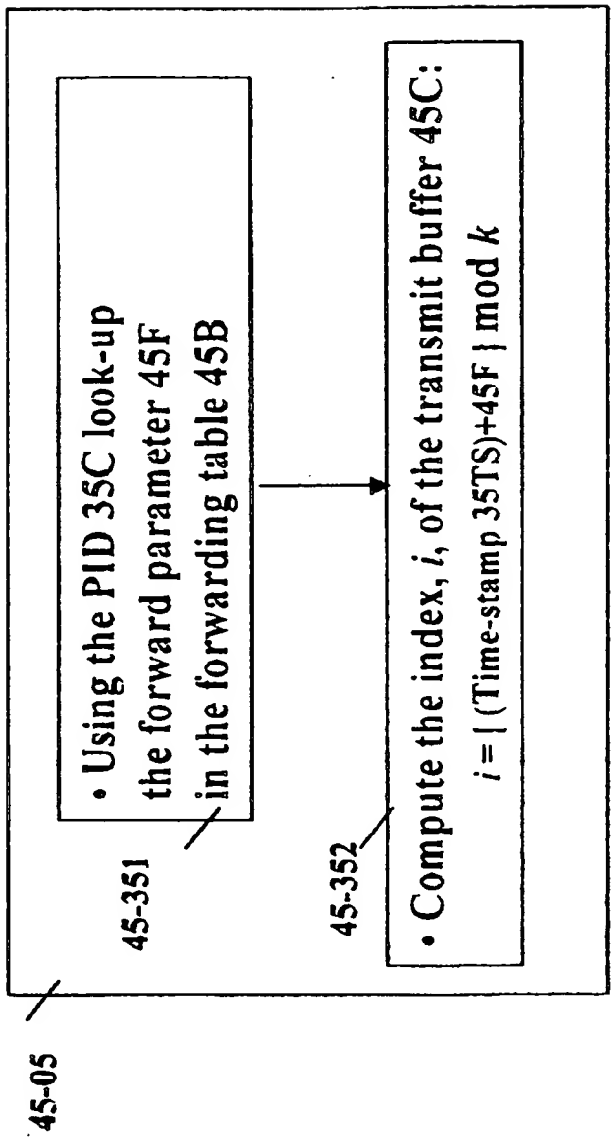

(A)
Input Port
41
30
30R
Resynchronization Buffer
37
To the Fabric
(B)
• Using the PID 35C look-up the delay bound uncertainty parameter 35DB (which is [Maximum delay - Minimum delay]) in the routing table 35B
35-151
35-152
• Delay the packet at the input port by: 35DB - |(CTR 002) - (Time-stamp 35TS)|

FIG. 62


45-05
45-351
• Using the PID 35C look-up the forward parameter 45F in the forwarding table 45B
45-352
• Compute the index, i, of the transmit buffer 45C:
i = | (Time-stamp 35TS)+45F | mod k

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/13311

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 12/66
US CL :370/352

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/352, 358, 370, 389, 391, 394, 395, 398

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 5,859,835 A (VARMA et al) 12 JANUARY 1999, Fig. 1, col. 1, lines 52-60, col. 2, lines 15-19, col. 5, lines 46-67, col. 6, lines 1-4, col. 11, lines 13-29 and col. 12, lines 16-64.	1-76
A	US 5,418,779 A (YEMINI et al) 23 MAY 1995, see entire document.	1-76

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 06 AUGUST 1999	Date of mailing of the international search report 29 OCT 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer CHI H. PHAM Telephone No. (703) 305-4378

Form PCT/ISA/210 (second sheet)(July 1992)*